**Source: Interdigital New York**

**Title: [FS\_AI4Media] pCR on update metadata**

**Spec: 3GPP TR 26.927 v0.9.0**

**Agenda item: 9.6**

**Document for: Agreement**

1. **Introduction**

Split point identifier can be defined as name, number or composition of these. Similarly, the identification of the tensors may be more generic than defined by names only.

Indeed, tensor names can contain very long strings of names, for example for the following *retinanet* tensors (

. e.g. /backbone/body/layer2/layer2.0/downsample/downsample.0/Conv\_output\_0, /head/regression\_head/conv/conv.0/conv.0.1\_2/InstanceNormalization\_output\_0.).

We propose to replace the tensor name to make it easier to process and exchange tensors. In the example above, we can substitute the original tensor name with shorter names (tensor\_1, tensor\_2) or with numerical index ( 1,2). These substitutions may be configured via a mapping table.

The contribution refines tensor identification to make it more generic. The proposal is to update clause 6.6 on metadata as follows:

* Redefine ‘tensor name’ to ‘tensor identifier’ to generalize the identification of a tensor.
* Introduce ‘tensor identifier type’ describing how the tensors are identified, for example, by a string (e.g. tensor1), a numerical value (e.g. 10, hash value), or by a composition of a string and a numerical value.
* Introduce ‘tensor mapping identifier’ as a unique identifier of a data model representation (e.g. list, table, structure) that maps tensor identifiers with original tensor names.

The tensor identifier type and the tensor mapping identifier can be defined and negotiated during the configuration stage. The tensor identifier can be exchanged during the configuration or the inference stage.

1. **Reason for Change**

Update the metadata section of the TR to refine the tensor identification.

1. **Proposal**

It is proposed to agree the following changes to 3GPP TR 26.927 v0.90.

\* \* \* beginning of Change 1 \* \* \* \*

## 6.6 Metadata

### 6.6.1 Introduction

Metadata for AI media services may include information describing AI models, inference requirements, endpoint capabilities (UE or network) and information more specific to the configuration, control and management of the basic AI service scenarios (AI model delivery, split AI/ML operation and distributed/federated learning).

NOTE: The delivery of the metadata described in this clause is not specified.

NOTE 2: How consistent and interoperable signalling of metadata (such as of existing framework mentioned in clause 6.4) may be ensured is FFS.

### 6.6.2 Common AI model information

AI model information metadata is used to describe the characteristics of AI models which may be used for an AI media service. This information may be common to all three AI service scenarios, and may be used in the selection of a suitable AI model by the UE or network, given an AI media service.

|  |  |  |  |
| --- | --- | --- | --- |
| **Metadata category** | **Metadata type** | **Definition** | **Metadata type description (Examples)** |
| **Model information** | **Model identifier** | An identifier for an AI model (or variants of it) specified for a certain AI media service. The identifier may be a name, a number, a combination thereof, a hash value. The identifier is defined during the configuration stage. | model\_1, model\_2 |
| **Number of parameters** | Total number of parameters in the neural network. | 11 million |
| **Model size** | The size of the AI model file in megabytes. | 40MB |
| **Input size** | The maximum size of the input data supported by the AI model in kilobytes. | 256 KB |
| **Output size** | The maximum size of the output data supported by the AI model in kilobytes. | 256 KB |
| **Accuracy** | The trained accuracy of the AI model as a percentage. | 85% |
| **Target inference latency** | The target inference latency specified for a given AI model in milliseconds. Such latency is measured between the input and output layers of the AI model at inference. This value is related to the service inference latency requirement of the service for which the AI model is provided, as well as the typical hardware capabilities of an entity performing the inference of the model. | 20ms |
| **Format/ framework** | The format or framework used to express the AI model, including its version number. | Pytorch 2.0 ONNX 1.15.0 |
| **Processing capabilities** | Estimated capabilities for processing the model including the computational power such as the computational cost (in FLOPS), the computational complexity (in MAC operations). It also includes the temporary memory to store model parameters.  | NPU 10TFLOPS, MEM 10GB |

### 6.6.3 AI model information for split AI/ML operations

AI model information metadata for split AI/ML operations is used to describe the characteristics of AI models for split inference service scenarios. This information may be used in the selection of a split point (from which a multiple may be predefined by the service provider for a certain AI media service). A trained model can be represented as a directed acyclic graph model represented by a collection of nodes interconnected with edges (e.g. ONNX). A split point may happened before or after a graph node identified by its name or a number.

|  |  |  |  |
| --- | --- | --- | --- |
| **Metadata category** | **Metadata type** | **Definition** | **Metadata type description (Examples)** |
| **Split model information** | **Split points** | The number of predefined split points at which a certain model can be divided into two for split inferencing. | 2 |
| **Split point information** | **Split point identifier** | An identifier of the split point in a description of a computing graph, may be generated by a neural network description language such as ONNX/NNEF. Identifiers must guarantee unique identification of a specific split point. | Nb:10, 75 Name: Layer\_10, |
| **Split point intermediate data size** | The size of the intermediate data resulting from the given split point, in kilobytes. Intermediate data size is typically dependent on the tensor size at the given split point. | 1086KB |
|  | **Split point number** | The number of the split point where the split occurs. The number may belong to set of identified numbers defined at the configuration stage.  | 10 |
| **Split point name** | The name of the split point where the split occurs. The name may belong to set of identified split point names defined at the configuration stage.  | conv2d\_1234 |
| **Split point flag** | An information on whether to consider the split point before the split point identifier or after. The convention on whether it is before or after may be defined at the configuration stage. | before, after |

### 6.6.4 Intermediate data information for split AI/ML operations

Intermediate data information identifies the structure of intermediate data output from a first endpoint that need to be retrieved to feed the inference of the second endpoint after transmission of the intermediate data over the network.

|  |  |  |  |
| --- | --- | --- | --- |
| **Metadata category** | **Metadata type** | **Definition** | **Metadata type description (Examples)** |
| **Intermediate data general information** | Tensor structure framework | The exact underlying tensor structure of the intermediate data tensors including the exact version of it. | PyTorch 2.0,TensorFlow v2.13.0, NumPy v1 .25 |
| Data direction  | This defines the direction of transmitted data, either uplink (from UE endpoint to network endpoint) or downlink (From a network endpoint to the UE endpoint). This information may be useful to configure an intermediate data delivery session | Upstream, Downstream |
| Global compression algorithm | Identifies a compression algorithm that can be applied to all the intermediate data tensors. For example, when the connectivity condition between the UE and the network is insufficient to transmit the original intermediate data, a compression algorithm may be applied. | NONE, FCM, SNAPPY, … |
|  | Tensor identifier type | The type of the tensor identifier describing how the tensors are identified. This parameter is defined during the configuration stage.  | String, Numerical value. |
|  | Tensor mapping identifier | A unique identifier of a data model representation (e.g. list, table, structure) that maps tensor identifiers with original tensor names. This parameter is defined during the configuration stage. | TensorTable12245, TensorList13 |
| **Intermediate data tensor information** | Tensor list | List of Tensors that composed the intermediate data | [tensor1, tensor2, tensor3, tensor4] |
|  |  | Tensor identifier | A unique identifier for the tensor. The identifier may be a name, an index of a tensor list or table, a combination thereof, a hash value.  | Tensor1, 10 |
|  |  | Tensor shape | Tensor shape is a tuple of positive integers, where the size of the tuple represents the dimension of the tensor, and each value represents the size in each dimension.  | [1,64,64,64].  |
|  |  | Tensor data type | The data type of each intermediate data tensor | Float32, int32 |
|  |  | Tensor compression algorithm | Identifies the compression algorithm(s) that can be applied to a particular tensor. The tensor compression algorithm supersedes the global compression algorithm when both are defined | NONE, FCM, SNAPPY, … |

\* \* \* End of Change 1 \* \* \* \*

\* \* \* Beginning of Change 2 \* \* \* \*

### 6.6.7 Compression metadata

#### 6.6.7.1 Compression settings for a split point configuration

##### 6.6.7.1.1 Introduction

The compression settings below identify:

- The candidate compression algorithms profiles to apply to intermediate data tensors.

- A split point compression characteristics (e.g. size reduction and performance metrics) for an association of compression algorithm profiles to intermediate data tensors.

##### 6.6.7.1.2 Compression algorithm profiles

A list of compression algorithm profiles includes a description, a unique identifier and associated parameter supported for intermediate data compression. A subset of compression algorithms profile can be negotiated and exchanged between endpoints regarding different endpoints capabilities.

|  |  |  |  |
| --- | --- | --- | --- |
| Metadata category | Metadata type | Definition | Metadata type description (Examples) |
| Compression algorithm profile | Compression algorithm profile list | List of compression algorithm profiles.  |  |
|  |  | Compression algorithm profile description | Identifies the compression algorithm(s) that can be applied to the intermediate data tensors. | NONE, Quantization, FC\_VCM, SNAPPYNeural Network Coding |
|  |  | Compression algorithm profile level identifier | Identifies the profile level of the compression algorithm.  | FCM high 5.1, FCM main 5.3, FCM 6.4NNC xxx 5.7.9, NNC yyy 5.8, NNC yyy 6.4  |
|  |  | Compression algorithm profile parameter set | List of compression parameters of the selected compression algorithm that fulfil the compression profile. | Param 1Param 2 e.g. Qp Quantization Parameter = -15 |

##### 6.6.7.1.3 Intermediate data tensors and associated compression profile and characteristics

This identifies information for associating individual or group of compression profiles to intermediate data tensors for a split point configuration. This includes split point information, expected split point compression characteristics associated to a compression profile.

|  |  |  |  |
| --- | --- | --- | --- |
| Metadata category | Metadata type | Definition | Metadata type description (Examples) |
| Split point information | Split point identifier | Key identifier of the split point clause §6.6.3 to associate value data below  | Nb:10, 75 Name: Layer\_10, |
| **Intermediate data general information** | Tensor identifier type | The type of the tensor identifier describing how the tensors are identified. This parameter is defined during the configuration stage.  | String, Numerical value. |
| Tensor mapping identifier | A unique identifier of a data model representation (e.g. list, table, structure) that maps tensor identifiers with original tensor names. This parameter is defined during the configuration stage. | TensorTable12245, TensorList13 |
| split point compression characteristics | Compressed intermediate data size  | The compressed intermediate data size. If no compression, the size is the baseline data size. | 12 Mbytes |
| Compressed intermediate data size ratio | The ratio gives an indication on how much the data has been reduced after apply compression. Example of representation of ratios: | ratio a) 5:1 or 5/1 means the size has been reduced from a factor 5ratio b) 0,8 means a reduction of 80% of the baseline size  |
| Compression performance metric value | The measured performance metric value depending on the performance metrics used, e.g. Map score, F1 score, accuracy. | F1 Score, Map score |
| Compression performance metric ratio | The ratio is calculated from the metric value obtained using compression divided by the value obtained without compression.for example, it may indicate how much accuracy has been affected  | Ratio 0.9 or 90% means a reduction performance accuracy of 10% |
| Intermediate data tensors associated to compression profile  | Tensor list | List of tensors or groups of tensors that composed intermediate data  | e.g. list of ONNX tensor names Tensor1, Tensor2 |
|  | Tensor group compression name | This identifies a group of tensors when each group is associated with a common compression profile. | Group 1, Group 2  |
|  | Tensor group compression type | This identifies the type for all tensors belonging to tensor group. | Float32  |
|  | Tensor compression granularity | This indicates if the compression profiles are applied to each tensor one by one (keyword is “tensor”), or to all the tensors of this group (keyword is “global”). | “tensor”, “global” |
|  | Tensor identifier | A unique identifier for the tensor. The identifier may be a name, an index of a tensor list or table, a combination thereof, a hash value.  | Tensor1, 10 |
|  | Tensor shape | Tensor shape output. The output tensor shape may be different from input and uncompressed tensor shape or may be transposed | [1,64,64,64].  |
|  | Tensor data type | The type of the tensor | Int32, Float32,  |
|  | Tensor compression algorithm profile identifier | Identifies the selected compression algorithm profile  | FCM high 5.1, FCM main 5.3, FCM 6.4NNC xxx 5.7.9, NNC yyy 5.8, NNC yyy 6.4 |

\* \* \* End of Change 2 \* \* \* \*