Frameworks: CNNs

- AlexNet
- VGG
- Inception
- ResNet

Inception M
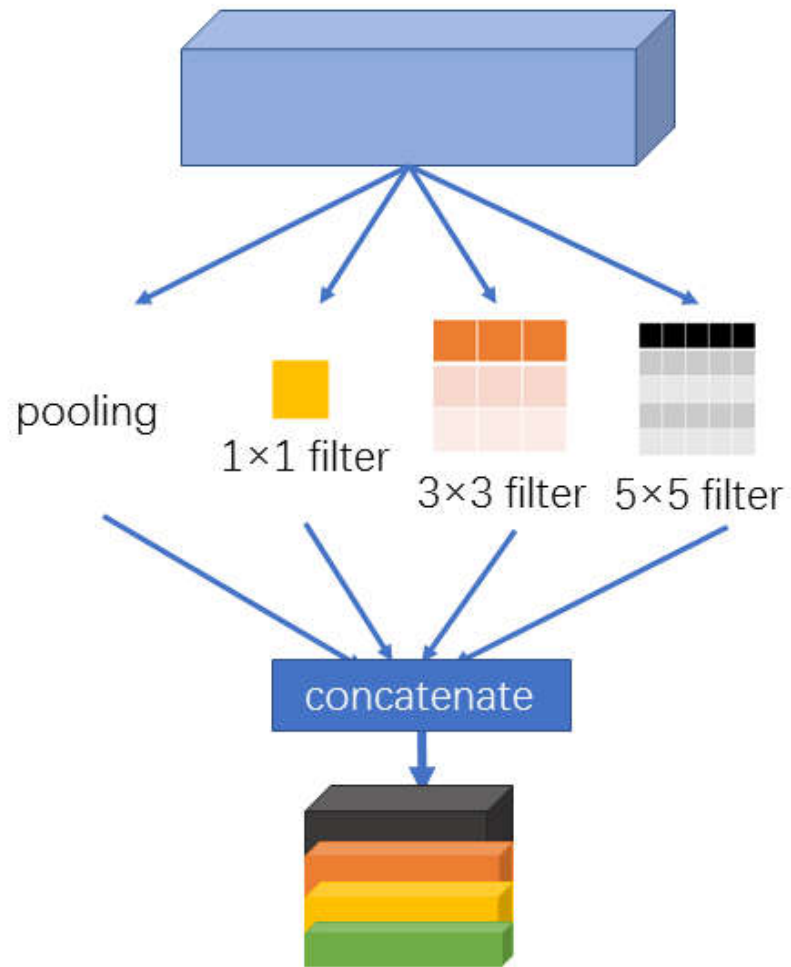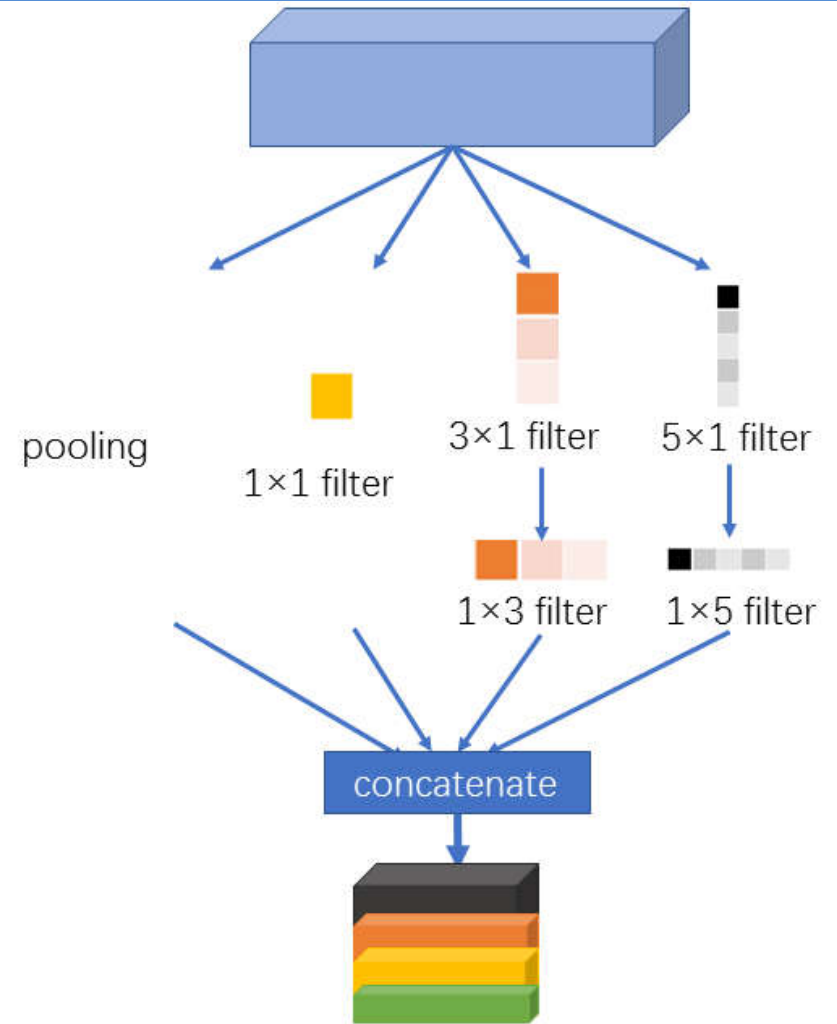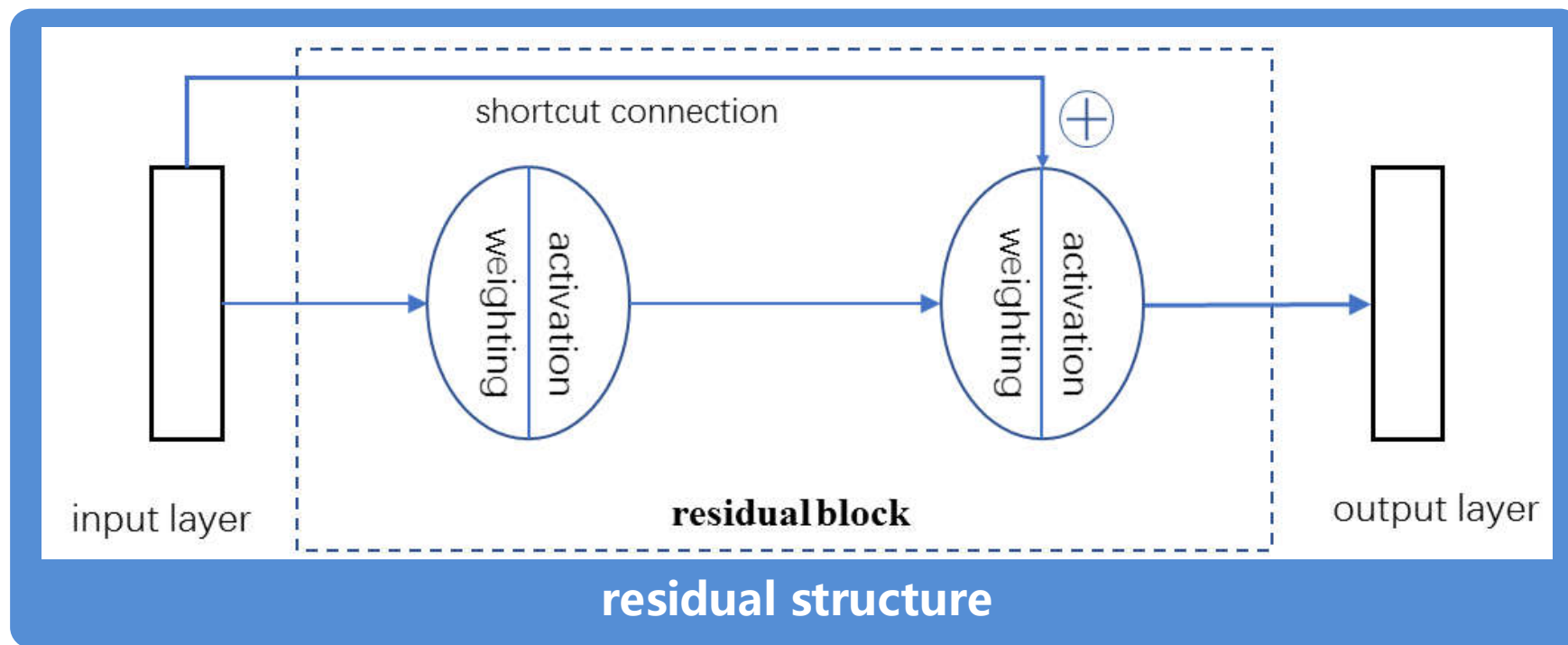
pooling

1×1 filter

3×3 filter 5×5 filter

concatenate

basic block

pooling

1×1 filter

3×1 filter 5×1 filter

1×3 filter 1×5 filter

concatenate

modified block

ResNet 🄼

shortcut connection

weighting | activation

weighting | activation

input layer

residual block

output layer

residual structure

Frameworks: RNNs [T]

- Basic RNN [C]
- GRU [C]
- LSTM [C]
- Bidirectional RNN [C]
- attention model [C]

Basic RNN 🟨Ⓜ

GRU 🔳
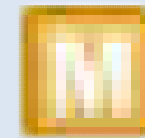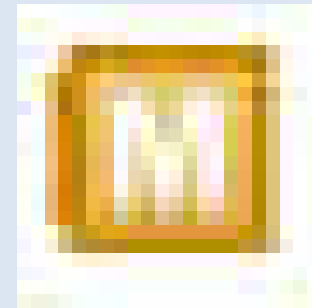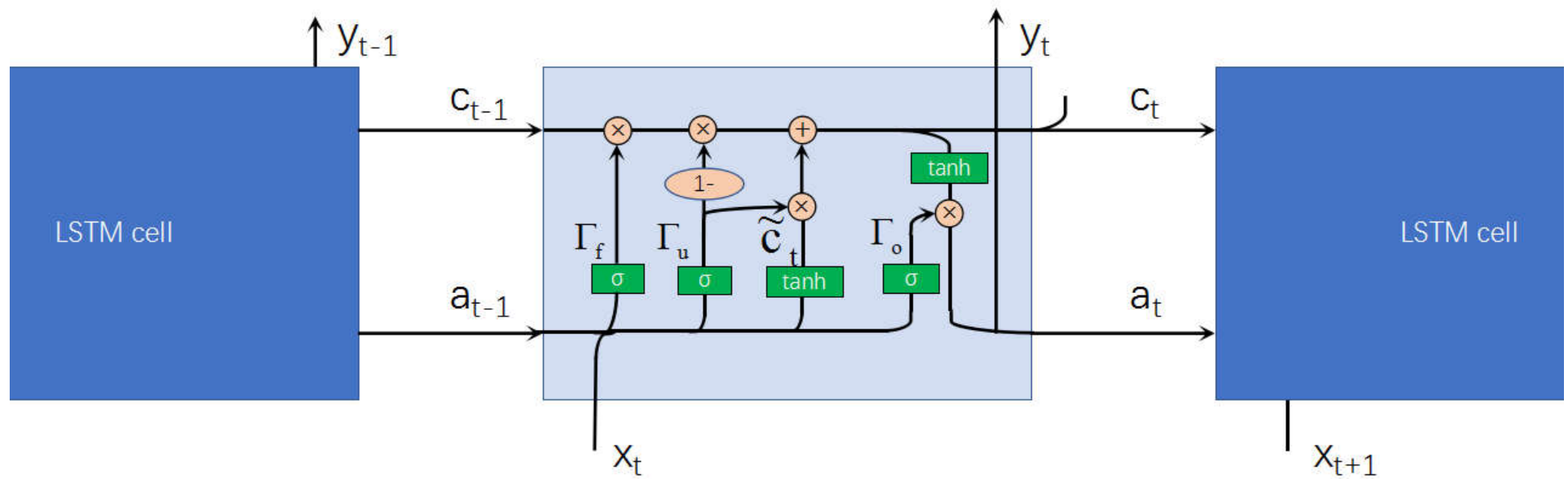
LSTM cell structure

Bidirectional RNN 🅼

y_{t-1} y_t y_{t+1}

x_{t-1} x_t x_{t+1}
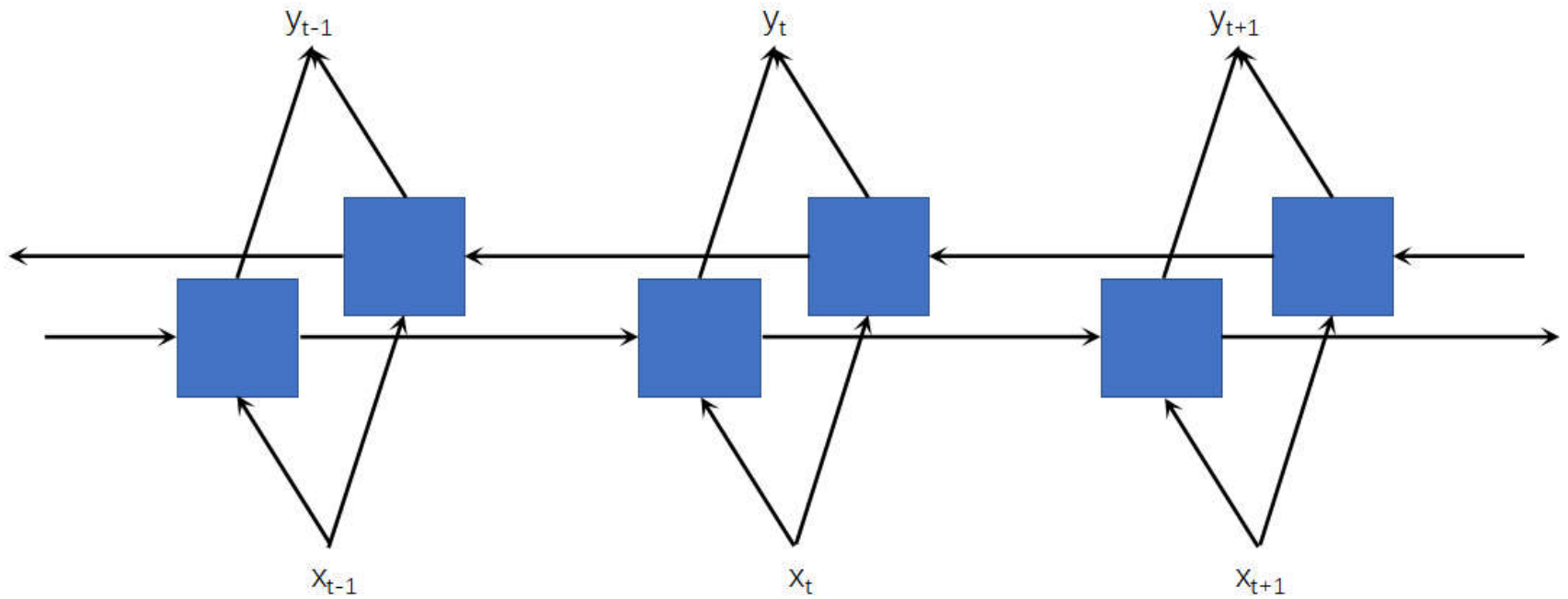
structure

attention model
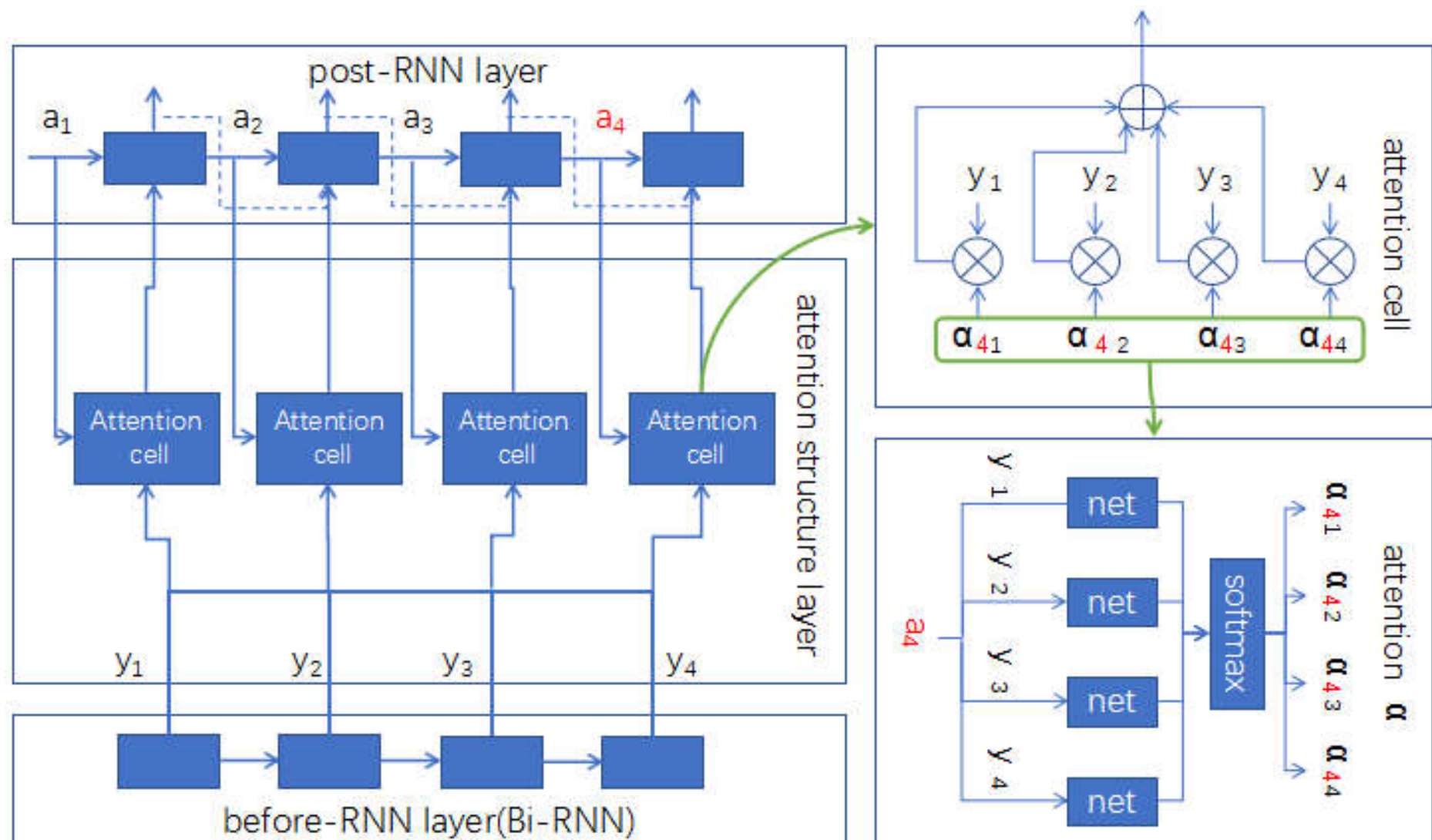
Cases: CNNs 🅣

- multi-classification
- face recognition 🅒
- style transfer generation 🅒
- objection recognition 🅒

RNNs

- sequence generation
- machine translation
- audio recognition

```
reference          verifing
feature            feature
  ↑                  ↑
┌──────────┐    ┌──────────┐
│          │    │          │
│  net_1   │net_1 and net_2│  net_2   │
│          │ are same │          │
│          │    │          │
└──────────┘    └──────────┘
  ↑                  ↑
reference          verifing
input              input
```

face recognition Ⓜ

- siamese net
  - 🟨 framework

- distance
  - 🟨 output

- triplet loss
  - 🟨 loss function

- binary classification
  - 🟨 output; loss function

$loss = \max(diffPosAndNeg, 0)$

$diffPosAndNeg = distPos + m \arg in - distNeg$

$output = \mathrm{sigmoid}(W * diffRefAndVerif + b)$

$$loss = \text{distance}(outputMidde\_G, outputMidde\_C)$$

$$loss = \text{distance}(matrixS\_G, matrixS\_S)$$

low

middle

high

style transfer generation

classification model source

context cost function

style cost function

whole cost function

$$loss = \alpha loss\_C + \beta loss\_S$$

classification model

outputMiddle_G

outputMiddle_C

classification model

graph_G

graph_C

classification model

outputMiddle_G

outputMiddle_S

classification model

matrixS_G

matrixS_S

graph_G

graph_S

$$matrixS = \begin{bmatrix} S_{11} & S_{1j} & S_{nn} \\ \cdots & & \cdots \\ S_{i1} & S_{ij} & S_{in} \\ \cdots & & \cdots \\ S_{n1} & S_{nj} & S_{nn} \end{bmatrix}$$

$$S_{ij} = \sum_{k}\sum_{t} a_{k,t}^{(i)} a_{k,t}^{(j)}$$

objection recognition 📄 Ⓜ

- object location
  - object localization
  - landmark finding
- window sliding
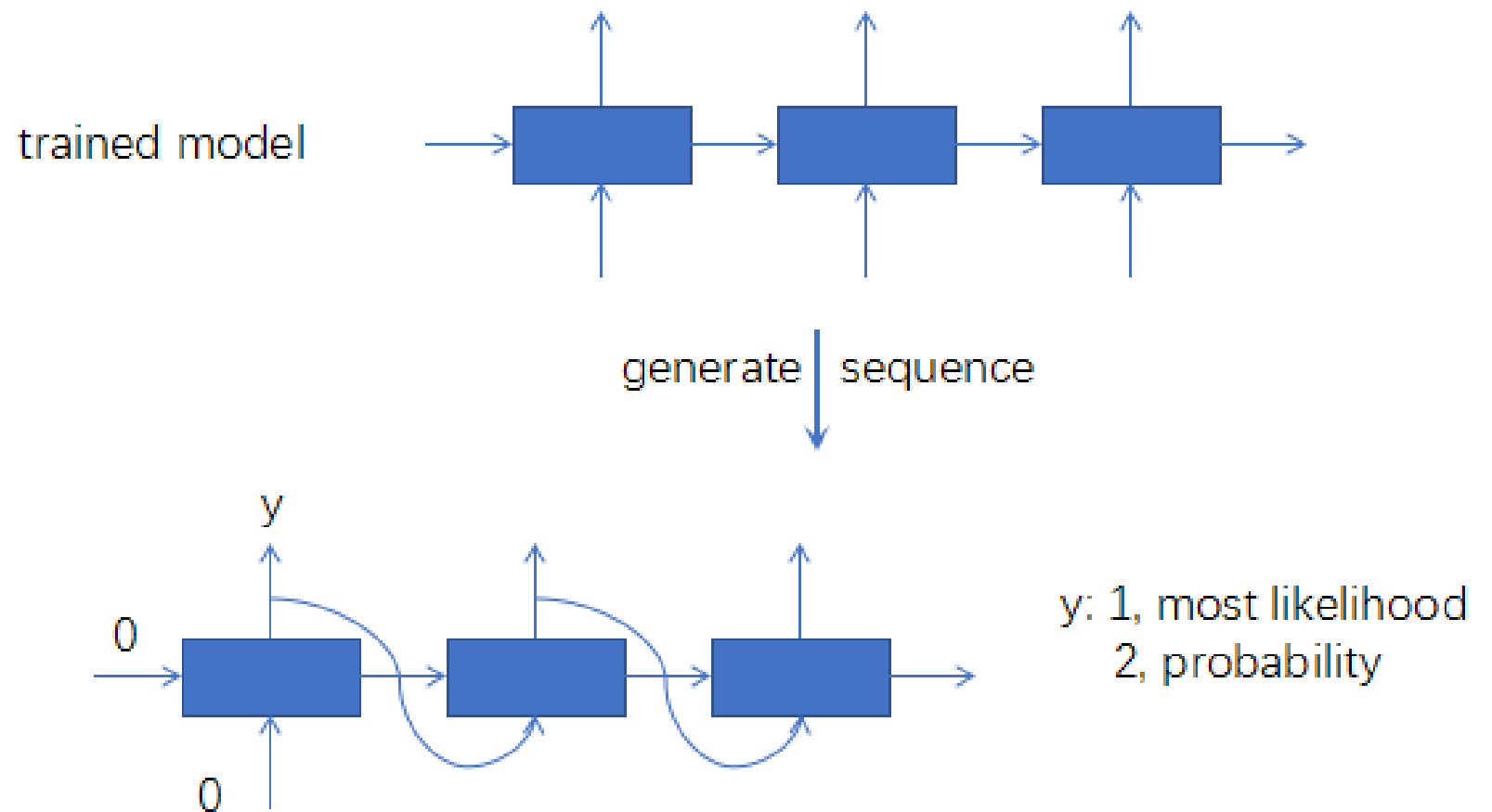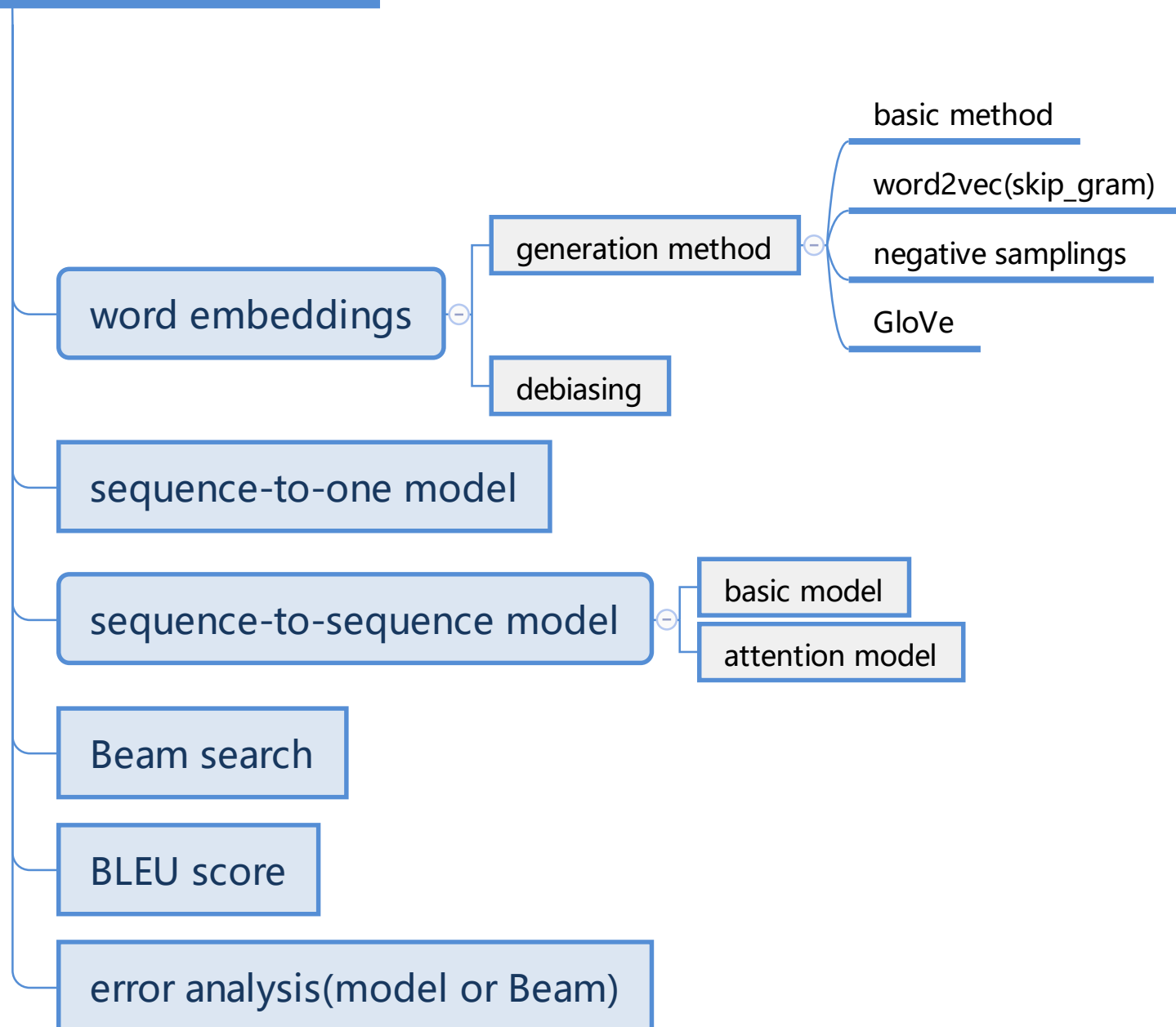- IoU
- non-max suppression
- anchor
- YOLO algorithm
- R-CNN algorithm

classifier

sequence generation

trained model

generate | sequence

y

0

0

y: 1, most likelihood
2, probability

machine translation 🅼

- **word embeddings**
  - generation method
    - basic method
    - word2vec(skip_gram)
    - negative samplings
    - GloVe
  - debiasing
- **sequence-to-one model**
- **sequence-to-sequence model**
  - basic model
  - attention model
- **Beam search**
- **BLEU score**
- **error analysis(model or Beam)**

audio recognition 🅜

- audio process
  - spectre transform
- audio labeling
  - delay label
  - repeat label
- trigger word
- sentence