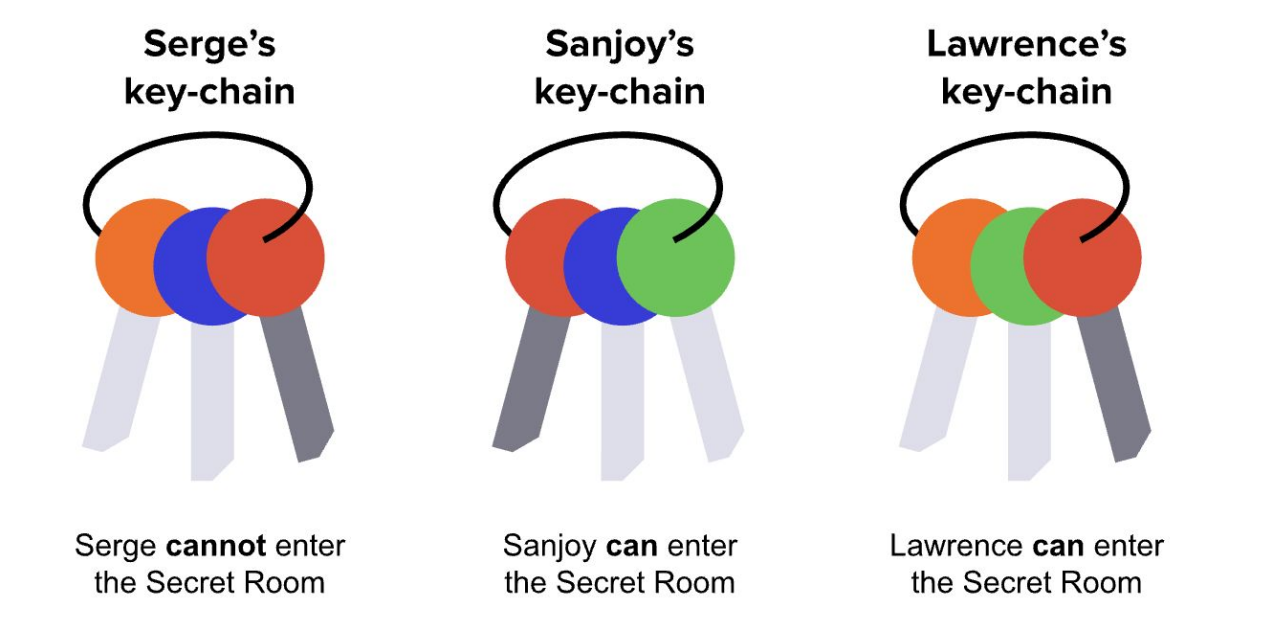
MIL

High Level

- Input data is grouped into labeled bags containing multiple instances
  - Labels are assigned at bag level
  - An entire bag is labeled as high-grade if it has 1 or more high-grade instances
  - Typical approaches:
    - Instance-level: classify each instance and aggregate to get bag-level classification
    - Embedding-level (preferred): map instances to low-level embeddings, aggregate the embeddings into a single, bag-level representation, and apply a classifier
- Terminology
  - Instances: individual data points (in our case, patches)
  - Bags: collection of data points
  - Labels: tagline for the bag (high-grade or benign)
- Why should we care?
  - Useful when labeling each instance is impractical/expensive
  - Especially useful for classifying medical image data

Key-chain analogy



Trying find the exact key that is common for all the "positive" keychains (green key)

Mathematical Foundation (1)

- Problem formulation
  - Bag of instances $X = \{x_1, x_2, ..., x_k\}$, each x has a label $y \in \{0,1\}$
  - Bag label $Y = \max\{y_k\}$
    - 1 if at least one instance is positive, else 0
  - During training, you don't know instance labels y_k
  - Use a permutation-invariant function:
    - Need to keep bag representation the same irrespective of order of instances within the bags
    - Punchline: can use average, max, or attention-based pooling
- Attention-based pooling
  - Use a weighted average of the low-level embeddings created during feature extraction
    - Max pooling only picks the most extreme patch, ignoring useful info from other patches
    - Average pooling treats every patch equally which includes irrelevant or incomplete patches

Mathematical Foundation (2)

- Attention-based pooling
  - Given embeddings H = {h_1,...., h_k}, the bag embedding is the sum

$$\mathbf{z} = \sum_{k=1}^{K} a_k \mathbf{h}_k,$$

where a_k is expressed as

$$a_k = \frac{\exp\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_k^\top\right)\}}{\sum_{j=1}^{K} \exp\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_j^\top\right)\}},$$

Mathematical Foundation (3)

$$a_k = \frac{\exp\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_k^\top\right)\}}{\sum_{j=1}^{K} \exp\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_j^\top\right)\}},$$

- Attention-based pooling
  - Each embedding transformed by a small neural network
    - V is a learned linear transformation, w is a learned weight
      - $V \in R^{LxM}$, $w \in R^{Lx1}$
      - L is size of the hidden layer and M is the size of the embedding (usually 512)
  - Tanh used so the model can capture nonlinear relationships
  - Rest is the softmax function

Explaining Attention (and Loss)

$$a_k = \frac{\exp\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_k^\top\right)\}}{\sum_{j=1}^{K} \exp\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_j^\top\right)\}},$$

Attention weights aren't trained directly, but learned indirectly through minimizing overall classification loss

- Use attention weights to aggregate into a single bag level prediction
- Then cross-entropy loss is computed between prediction and true bag label
- The loss is backpropagated through the classifier to calculate the gradients and update V, w, CNN, and the final classifier weights

h_k – the patch embedding for k-th instance of bag (output from patch classifier – a 1D vector of shape [512])

V – weight matrix in attention MLP, learnable parameter – a matrix of shape [128, 512]

w – weight vector in attention MLP, learnable parameter – a 1D vector of shape [128]

Tanh – nonlinear activation function, to capture nonlinearity into the model

a_k – Attention score for patch k (scalar value)

exp() – softmax function so that all attention weights $a_k \in (0,1)$

Mathematical Foundation (4)

- Gated attention mechanism

$$a_k = \frac{\exp\{\mathbf{w}^\top \left(\tanh\left(\mathbf{V}\mathbf{h}_k^\top\right) \odot \mathrm{sigm}\left(\mathbf{U}\mathbf{h}_k^\top\right)\right)\}}{\sum_{j=1}^{K} \exp\{\mathbf{w}^\top \left(\tanh\left(\mathbf{V}\mathbf{h}_j^\top\right) \odot \mathrm{sigm}\left(\mathbf{U}\mathbf{h}_j^\top\right)\right)\}},$$

  - Motivation: tanh is linear over [-1,1]
    - Gated attention adds the sigmoid function to introduce an additional learnable nonlinearity
    - $U \in R^{LxM}$ is an additional learned parameter, $\odot$ is element-wise multiplication
  - In theory, adds more flexibility to MIL