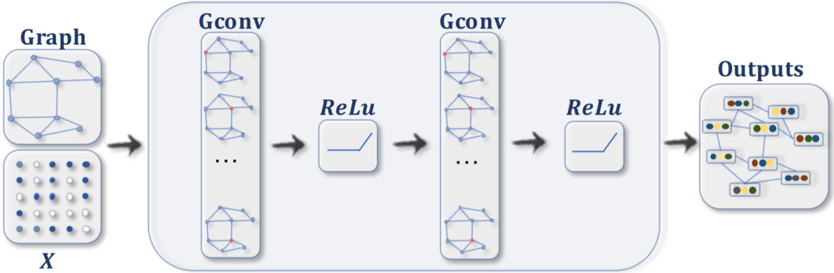
GCNN

Медведев Алексей Владимирович

МГУ имени М. В. Ломоносова, факультет ВМК, кафедра ММП

Зачем?

- Данные представлены в виде графа (например молекулы).
- Можно обучать классификаторы для multilabeling.
- Предсказание и распознавание действий.



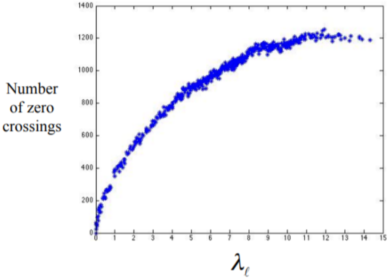
Спектральные сети

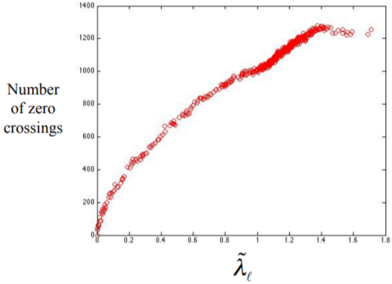
$$L = D - A \,;\, L = I_N - D^{-0.5} A D^{-0.5}$$

$$\hat{f}(\lambda_l) = \sum_{i=1}^{N} f(i) u_l(i) \,;\, f(i) = \sum_{l=0}^{N-1} \hat{f}(\lambda_l) u_l(i)$$

$$\mathcal{F}(x) = U^T x \,;\, \mathcal{F}^{-1}(x) = U \hat{x}$$

Спектральные сети





$$\mathcal{Z}_{\mathcal{G}}(f) := e = (i,j) \in \mathcal{E} : f(i)f(j) < 0$$

Спектральные сети

Свертка

$$x *_G g = \mathcal{F}^{-1}(\mathcal{F}(x) \odot \mathcal{F}(g)) = U(U^T x \odot U^T g) =$$

$$\{g_\theta = diag(U^T g)\} = U g_\theta U^T x$$

T. N. Kipf and M. Welling 2017 ICLR

Чебышев

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$$

$$T_0(x) = 1 \; ; \; T_1(x) = x$$

$$x *_G g \approx \sum_{k=0}^{K} \theta_k T_k(L) x$$

GCN

$$x *_G g \approx \theta_0 x + \theta_1 (L - I_N) x = \theta_0 x - \theta_1 D^{-0.5} A D^{-0.5} x$$

$$x *_G g \approx \theta (I_N + D^{0.5} A D^{-0.5}) x$$

T. N. Kipf and M. Welling 2017 ICLR

Renormalization trick

$$I_N + D^{-0.5} A D^{-0.5} \rightarrow \tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5}$$

$$\tilde{A} = A + I_N \, ; \tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$$

$$H^{(l+1)} = \sigma(\tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5} H^{(l)} W^{(l)})$$

T. N. Kipf and M. Welling 2017 ICLR

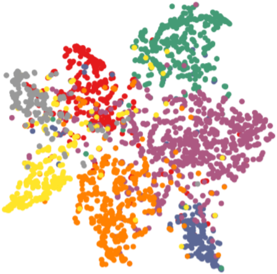
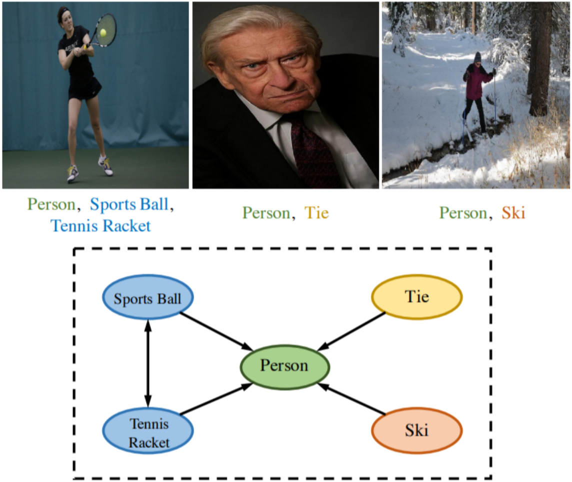


Рис.: Cora dataset t-SNE

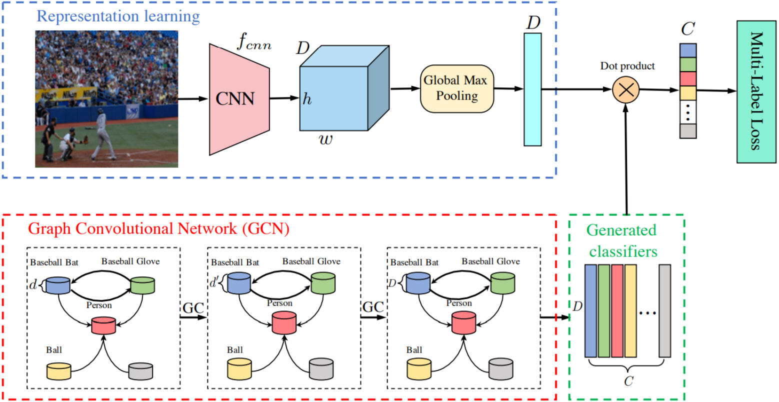
Multi-Label Image Recognition with Graph Convolutional Networks

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo 2019 CVPR



Multi-Label Image Recognition with Graph Convolutional Networks

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo 2019 CVPR



Adjacency matrix binary

$$P_i = M_i / N_i$$

$$A_{ij} = \begin{cases} 0, & \text{if } P_{ij} < \tau \\ 1, & \text{else} \end{cases}$$

Adjacency matrix reweighted

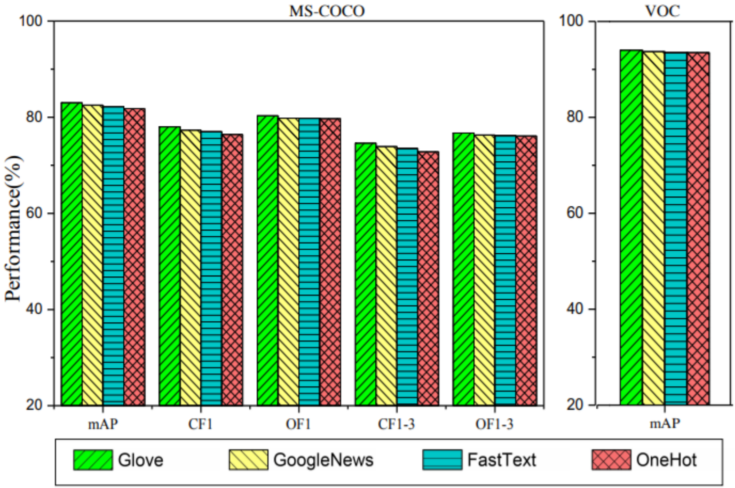
$$A_{ij} = \begin{cases} p / \sum_{j=1,\, j \neq i}^{C} A_{ij} & \text{if } i \neq j \\ 1 - p, & \text{else} \end{cases}$$

```python
def gen_A(num_classes, t, adj_file):
    import pickle
    result = pickle.load(open(adj_file, 'rb'))
    _adj = result['adj']
    _nums = result['nums']
    _nums = _nums[:, np.newaxis]
    _adj = _adj / _nums
    _adj[_adj < t] = 0
    _adj[_adj >= t] = 1
    _adj = _adj * 0.25 / (_adj.sum(0, keepdims=True) + 1e-6)
    _adj = _adj + np.identity(num_classes, np.int)
    return _adj

def gen_adj(A):
    D = torch.pow(A.sum(1).float(), -0.5)
    D = torch.diag(D)
    adj = torch.matmul(torch.matmul(A, D).t(), D)
    return adj
```

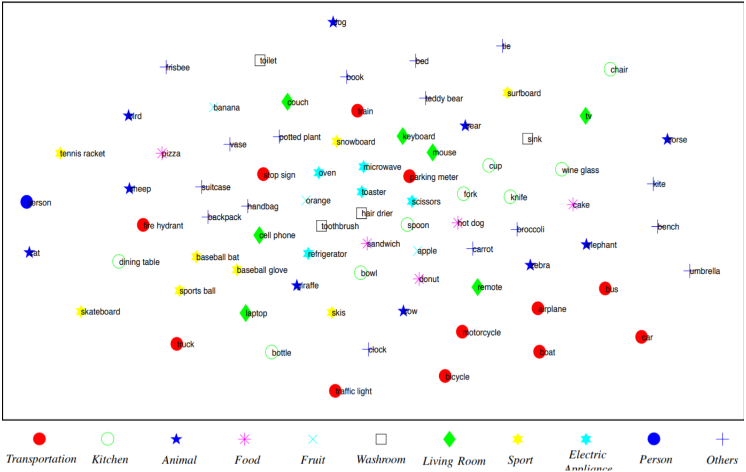
Multi-Label Image Recognition with Graph Convolutional Networks

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo 2019 CVPR



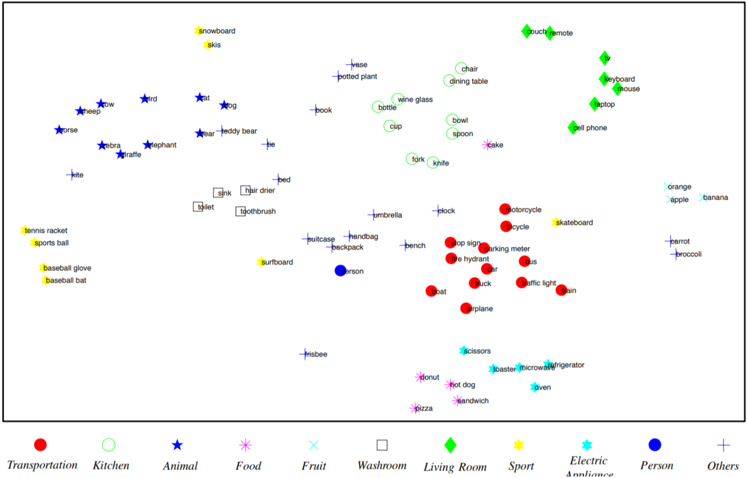
Multi-Label Image Recognition with Graph Convolutional Networks

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo 2019 CVPR



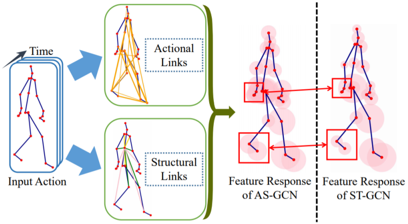
Multi-Label Image Recognition with Graph Convolutional Networks

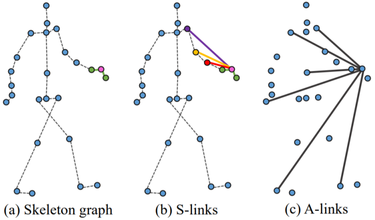
Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, Yanwen Guo 2019 CVPR

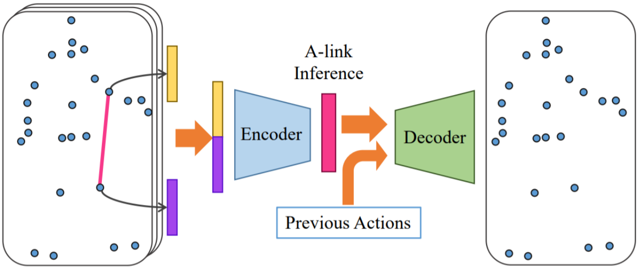


Actional-Structural Graph Convolutional Networks for Skeleton-based Action Recognition

Maosen Li , Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian 2019 CVPR







Input Action Sequence

Future Action

Encoder part

link features:$\mathbf{Q}_{ij}^{k+1} = f_e^k(f_v^k(p_i^k) \oplus f_v^k(p_j^k))$,
joint features: $p_i^{k+1} = \mathcal{F}(Q^{k+1}) \oplus p_i^k$

$$\mathcal{A}_{i,j,:} = \text{softmax}\left(\frac{Q_{i,j}^K + r}{\tau}\right) \in \mathbb{R}^C$$

Decoder part

link features:$\mathbf{Q}_{ij}^{t} = \sum_{c=1}^{C} \mathcal{A}_{i,j,c} f_e^c(f_v^c(x_i^t) \oplus f_v^c(x_j^t))$,
joint features: $p_i^t = \mathcal{F}(Q_{i,:}^t) \oplus x_i^t$
hidden state: $S_i^{t+1} = GRU(S_i^t, p_i^t)$
expectation of position:$\hat{\mu}_i^{t+1} = f_out(S_i^{t+1}) \in \mathbb{R}^3$

Loss

$$\mathcal{L}_{AIM}(\mathcal{A}) = -\sum_{i=1}^{n}\sum_{t=2}^{T}\frac{\|x_i^t - \hat{\mu}_i^t\|^2}{2\sigma^2} + \sum_{c=1}^{C}\log\frac{\mathcal{A}_{:,:,c}}{\mathcal{A}^0_{:,:,c}}$$

Prior

$$\forall i,j,\ \mathcal{A}_{i,j,0}\sum_{c=1}^{C}\mathcal{A}_{i,j,c} = 1,$$

$$\mathcal{A}^0_{i,j,0} = P_0,\ \mathcal{A}^0_{i,j,c} = P_0/C,\ \hat{A}^{(c)}_{act} = D^{(c)^{-1}}_{act}A^{(c)}_{act}$$

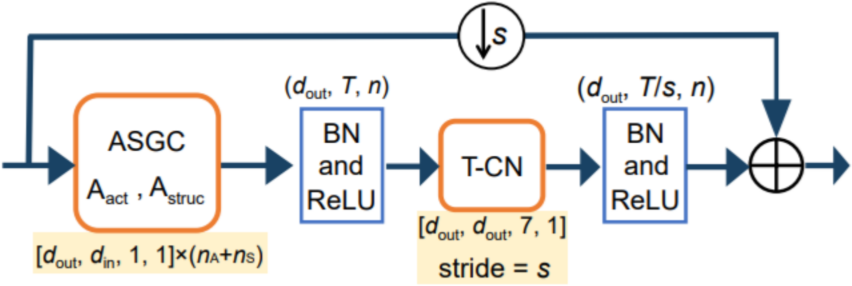
AGC

$$X_act = AGC(X_{in}) = \sum_{c=1}^{C}\hat{A}^{(c)}_{act}X_{in}W^{(c)}_{act} \in \mathbb{R}^{n\times d_{out}}$$

Structural Convolution

$$\hat{A} = D^{-1}A$$

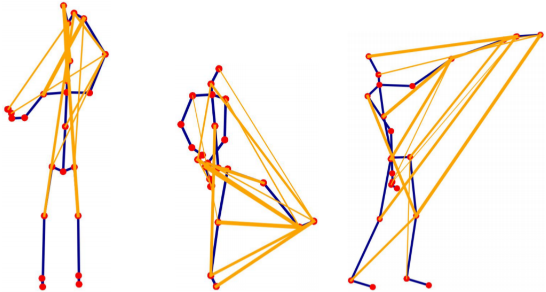
$$X_struc = SGC(X_in) = \sum_{l=1}^{L} \sum_{p \in \mathcal{P}} M_{struc}^{(p,l)} \odot \hat{A}^{(p)l} X_{in} W_{struc}^{(p,l)} \in \mathbb{R}^{n \times d_{out}}$$



$$\mathcal{L}_{recog} = -y^T \log(\hat{y}), \; \mathcal{L}_{predict} = \frac{1}{ndT'} \sum_{i}^{nd} \sum_{t=1}^{T'} \| \hat{\mathcal{X}}_{i,:,t} - \mathcal{X}_{i,:,t} \|_2^2$$

Actional-Structural Graph Convolutional Networks for Skeleton-based Action Recognition

Maosen Li , Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian 2019 CVPR



(a) Hand waving (b) Kick something (c) Taking a selfie

C	1	2	3	4	5
Acc	84.6%	86.5%	**86.8%**	85.8%	83.3%
P_0	0.99	0.95	0.50	0.20	0.00
Acc	86.0%	**86.8%**	84.3%	82.7%	81.1%

Actional-Structural Graph Convolutional Networks for Skeleton-based Action Recognition

Maosen Li , Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian 2019 CVPR