深度学习论文翻译作业

李佳政

201828013229075

计算技术研究所

DKN: Deep Knowledge-Aware Networkfor News Recommendation

- 基于知识图谱的新闻推荐网络框架。
- 由于新闻领域的实体很多，所以融合知识后，对新闻内容的解析更加完整。
- 知识图谱主要由实体和关系构成，将新闻的标题和知识库实体进行对齐，增强新闻的内容。
- DKN选用了多个特征：词向量，实体向量，知识图谱上下文实体向量作为输入表示。
- 使用一个卷积操作对齐，融合了语义和知识的表示。
- 最后使用注意力机制，对和用户历史兴趣的多主题进行建模预测。

DKN: Deep Knowledge-Aware Networkfor News Recommendation

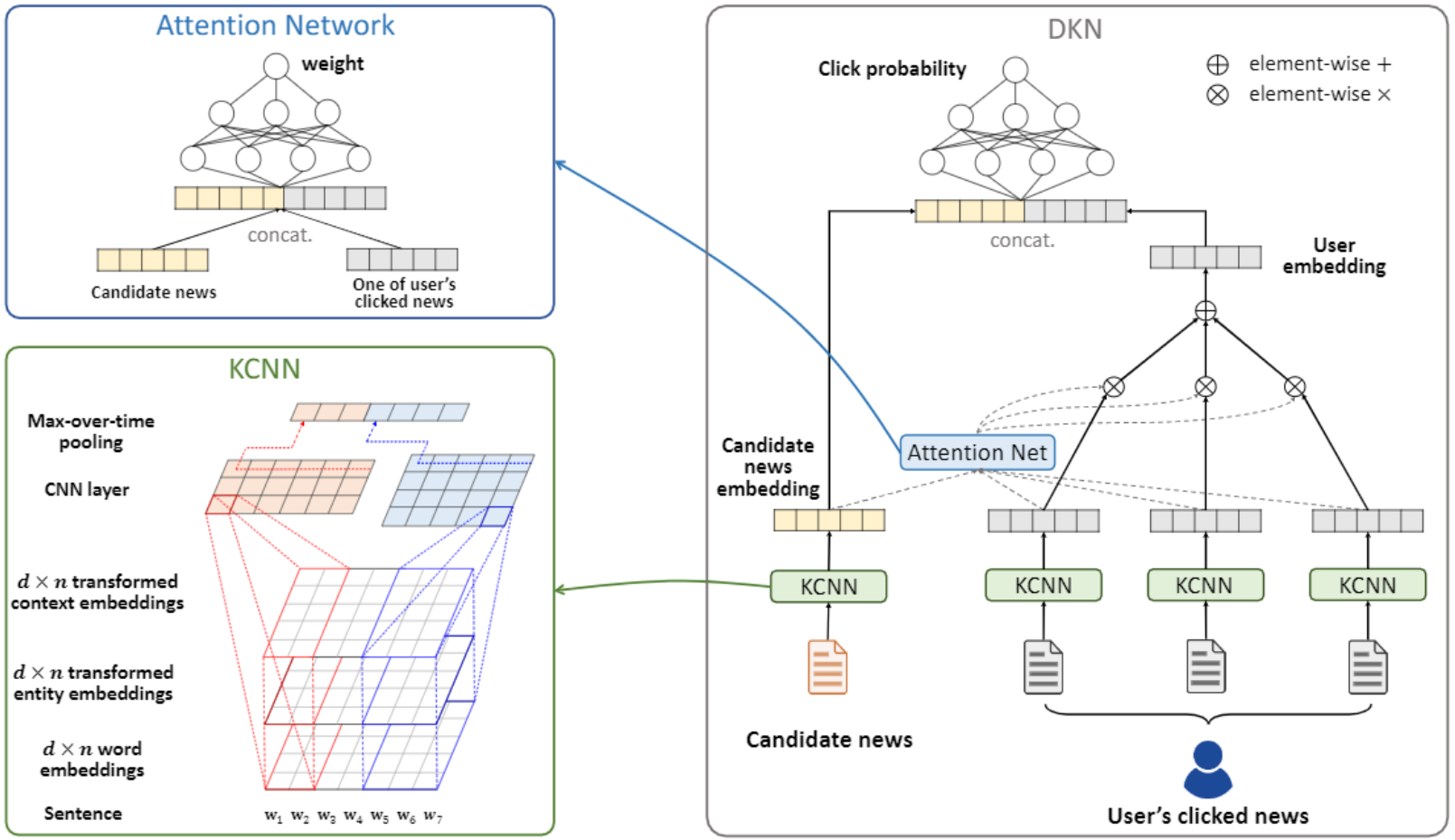
- 基于
- 由于 更加
- 知识 行对
- DKN 向量
- 使用
- 最后

的解析

实体进

文实体

模预测。



DKN: Deep Knowledge-Aware Networkfor News Recommendation

- 对知识图谱进行向量化，有transE，transH，transR和transD模型。
- 上下文实体：与新闻标题中实体有直接相连的关系的实体，可以帮助定位具体的实体的主题位置。
- 基于注意力的用户兴趣抽取：根据用户历史点击的数据，进行注意力机制评分和归一化，可以对同一个用户多个兴趣主题进行有效建模。
- DKN效果取得了最好的推荐结果。

DKN: Deep Knowledge-Aware Networkfor News Recommendation

- 对知识图
- 上下文实帮助定位
- 基于注意意力机制效建模。

Table 2: Comparison of different models.

Models*	F1	AUC	*p*-value**
DKN	**68.9 ± 1.5**	**65.9 ± 1.2**	–
LibFM	61.8 ± 2.1 (-10.3%)	59.7 ± 1.8 (-9.4%)	$< 10^{-3}$
LibFM(-)	61.1 ± 1.9 (-11.3%)	58.9 ± 1.7 (-10.6%)	$< 10^{-3}$
KPCNN	67.0 ± 1.6 (-2.8%)	64.2 ± 1.4 (-2.6%)	0.098
KPCNN(-)	65.8 ± 1.4 (-4.5%)	63.1 ± 1.5 (-4.2%)	0.036
DSSM	66.7 ± 1.8 (-3.2%)	63.6 ± 2.0 (-3.5%)	0.063
DSSM(-)	66.1 ± 1.6 (-4.1%)	63.2 ± 1.8 (-4.1%)	0.045
DeepWide	66.0 ±1.2 (-4.2%)	63.3 ± 1.5 (-3.9%)	0.039
DeepWide(-)	63.7 ± 0.9 (-7.5%)	61.5 ± 1.1 (-6.7%)	0.004
DeepFM	63.8 ± 1.5 (-7.4%)	61.2 ± 2.3 (-7.1%)	0.014
DeepFM(-)	64.0 ± 1.9 (-7.1%)	61.1 ± 1.8 (-7.3%)	0.007
YouTubeNet	65.5 ± 1.2 (-4.9%)	63.0 ± 1.4 (-4.4%)	0.025
YouTubeNet(-)	65.1 ± 0.7 (-5.5%)	62.1 ± 1.3 (-5.8%)	0.011
DMF	57.2 ± 1.2 (-17.0%)	55.3 ± 1.0 (-16.1%)	$< 10^{-3}$

和transD模型。

的实体，可以

数据，进行注
趣主题进行有

Unsupervised Image-to-Image Translation Networks

- Deepfakes的启发式论文。
- 根据图像边缘的联合分布建模，认为可以通过穷举的方式来完成，不做其他的假设。为了解决图像翻译问题，建立了一个共享隐空间作为中间转换层。
- 基于VAE变分自编码器和GAN生成对抗网络。
- 共包含6个部分，2个VAE，2个生成网络，2个判别网络。
- 权重共享，2个VAE的最后一层共享参数。生成网络的前几层共享参数。
- 共享隐空间，隐编码是还原两张不想关图片的纽带。每个编码后的图片输入到两个G中，每次D能学到两个样例，判断是G生成的哪个输入图像。

Unsupervised Image-to-Image Translation Networks

- 目标函数

$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}_{\text{VAE}_1}(E_1,G_1) + \mathcal{L}_{\text{GAN}_1}(E_2,G_1,D_1) + \mathcal{L}_{\text{CC}_1}(E_1,G_1,E_2,G_2)$$
$$\mathcal{L}_{\text{VAE}_2}(E_2,G_2) + \mathcal{L}_{\text{GAN}_2}(E_1,G_2,D_2) + \mathcal{L}_{\text{CC}_2}(E_2,G_2,E_1,G_1).$$

- 自编码器损失

$$\mathcal{L}_{\text{VAE}_1}(E_1,G_1) = \lambda_1 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1\sim q_1(z_1|x_1)}[\log p_{G_1}(x_1|z_1)]$$
$$\mathcal{L}_{\text{VAE}_2}(E_2,G_2) = \lambda_1 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_2\sim q_2(z_2|x_2)}[\log p_{G_2}(x_2|z_2)].$$

- GAN损失

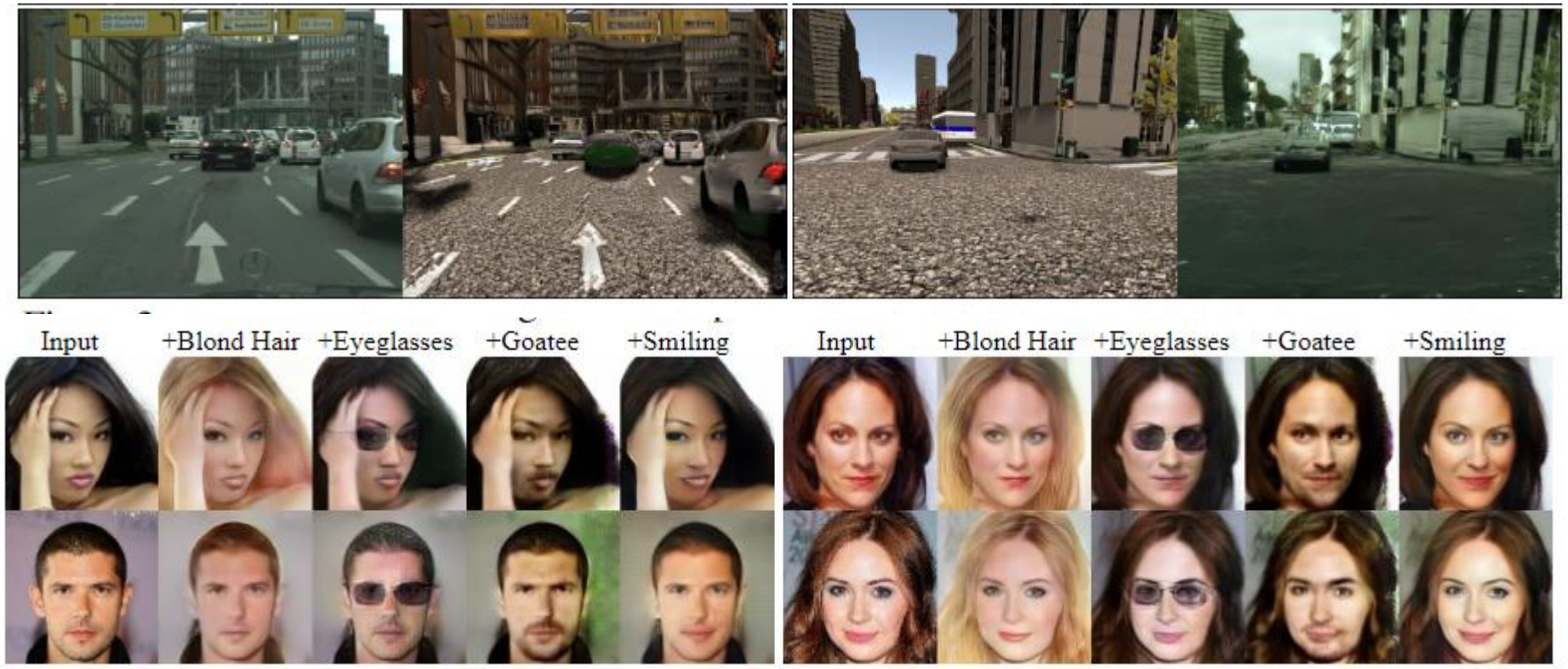
$$\mathcal{L}_{\text{GAN}_1}(E_2,G_1,D_1) = \lambda_0 \mathbb{E}_{x_1\sim P_{\mathcal{X}_1}}[\log D_1(x_1)] + \lambda_0 \mathbb{E}_{z_2\sim q_2(z_2|x_2)}[\log(1 - D_1(G_1(z_2)))]$$
$$\mathcal{L}_{\text{GAN}_2}(E_1,G_2,D_2) = \lambda_0 \mathbb{E}_{x_2\sim P_{\mathcal{X}_2}}[\log D_2(x_2)] + \lambda_0 \mathbb{E}_{z_1\sim q_1(z_1|x_1)}[\log(1 - D_2(G_2(z_1)))].$$

- 一致性限制

$$\mathcal{L}_{\text{CC}_1}(E_1,G_1,E_2,G_2) = \lambda_3 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) + \lambda_3 \text{KL}(q_2(z_2|x_1^{1\to 2}))||p_\eta(z)) -$$
$$\lambda_4 \mathbb{E}_{z_2\sim q_2(z_2|x_1^{1\to 2})}[\log p_{G_1}(x_1|z_2)]$$
$$\mathcal{L}_{\text{CC}_2}(E_2,G_2,E_1,G_1) = \lambda_3 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) + \lambda_3 \text{KL}(q_1(z_1|x_2^{2\to 1}))||p_\eta(z)) -$$
$$\lambda_4 \mathbb{E}_{z_1\sim q_1(z_1|x_2^{2\to 1})}[\log p_{G_2}(x_2|z_1)].$$

Unsupervised Image-to-Image Translation Networks

- 实验结果，互相翻译



Distilling the Knowledge in a Neural Network

- 在机器学习中，许多集成方法就是训练多个模型然后平均他们的预测结果进行输出，如果对于大型神经网络，那么时间复杂度是很高的，所以如何把多个模型集成到一个模型中进行预测，就是本文研究的一个问题。
- 这个训练方法可以并行训练。
- 模型可以提供一个soft label，而不是真实标签那样错误的为0，使用退火算法。
- 使用两个目标函数。1）与软标签的交叉熵

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

2）与真实标签的交叉熵

Distilling the Knowledge in a Neural Network

- 简化:

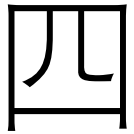
$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right) \qquad \frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z_i - v_i)$$

- 损失可以表示原始复杂模型对于数据的拟合程度，可能会包含噪声。
- 语音识别效果

System	Test Frame Accuracy	WER
Baseline	58.9%	10.9%
10xEnsemble	61.1%	10.7%
Distilled Single model	60.8%	10.7%

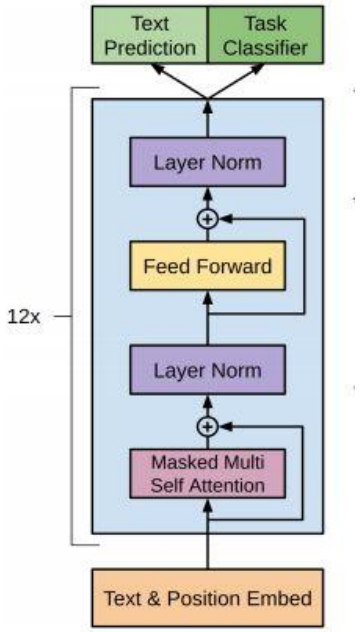
- Soft label可以阻止过拟合。

System & training set	Train Frame Accuracy	Test Frame Accuracy
Baseline (100% of training set)	63.4%	58.9%
Baseline (3% of training set)	67.3%	44.5%
Soft Targets (3% of training set)	65.4%	57.0%



Improving Language Understanding by Generative Pre-Training

- openAI 的GPT模型，最早采用双向transformer来建立语言模型的工作。以往的模型如ELMo,BiLSTM都是使用RNN。
- 模型的结构不适用于所有的任务，需要针对任务来调整模型的结构。
- Pretrained+fine-tuning
- 模型的能力很大程度来源于transformer。
- 主要用于自然语言生成任务。



五

Language Models are Unsupervised Multitask Learners

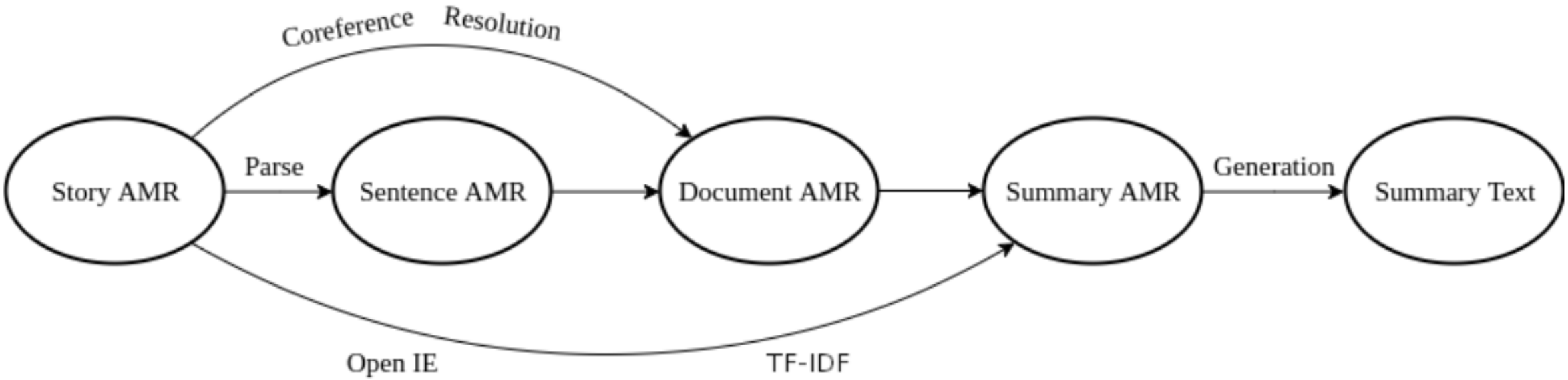
- openAI在BERT后提出的一个模型**GPT-2**，与前一代的区别主要是模型深度不同，参数量15亿。主要针对的任务是BERT无能为力的文本生成方面。
- Zero-shot，该模型证明了在语料充足的情况下，语言模型能够学习到一些其他领域的知识而不必专门训练。
- 基于任务的语言模型。
- 与BERT同样都是暴力的模型，由于参数量过大，语料不足，underfit。
- 生成的文本质量比较随机，没有推理能力，神经网络也许不是文本生成的好模型。

六

Unsupervised Semantic Abstractive Summarization

- 无监督：没有训练过程。有标签作为评价指标。
- 语义：将文档以AMR图的形式重新组织，人可理解的方式。
- 摘要：根据AMR图抽取出重要的事件关系，重新组织成摘要文本。
- 摘要核心流程包括：1.找到重要的实体/事件 2. 找到实体之间的重要关系 3. 提取出信息后，重新组织文本。
- 具体流程：1. 文档转换为AMR图 2. AMR图简化 3. 根据TF-IDF，关系抽取形成摘要。
- 1. 如何找到重要的节点？根据词的TF-IDF值
- 2. 如何找到重要的关系？启发式方法，与关键节点共同出现，并且离根节点更近。
- 3.如何应用AMR子图？找到对应的句子后抽取三元组。三元组可以被认为是结构化的数据，能用来生成文本。
- AMR抽取器的设计采用encoder-decoder架构。

Unsupervised Semantic Abstractive Summarization



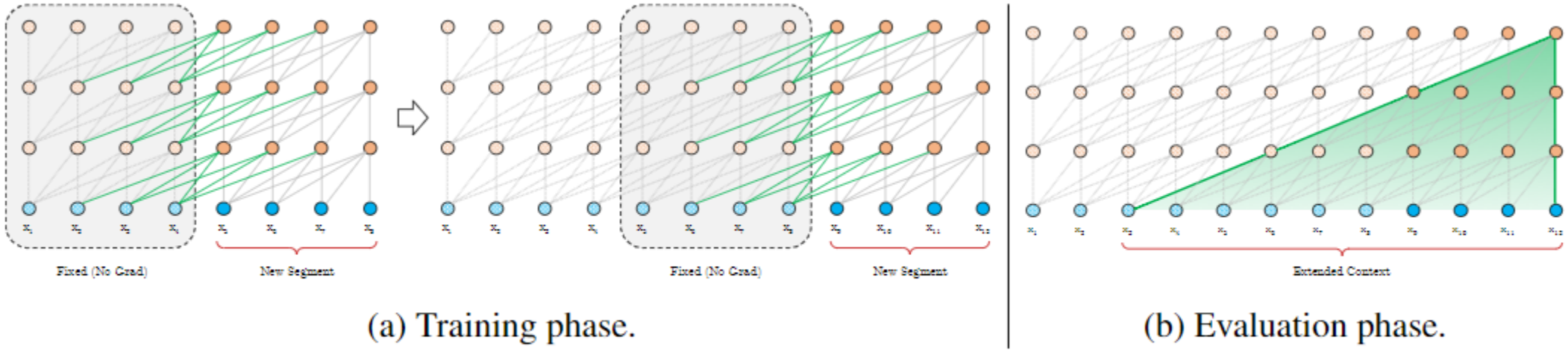
七

Transformer-XL: Attentive Language Models Beyond A Fixed-Length Context

- 虽然Transformer有很好的并行化的内力，但是self-attention的方法对于长文本来说内存消耗太大，所以传统的Vanilla方法无法依赖距离长的信息，训练的时候各个fragment是分开训练的，在测试的时候，每次移动一步来计算，所以速度比较慢。
- Transformer-XL主要针对这个问题进行优化，XL代表extra long。主要思想就是每个fragment依赖于前一个fragment的隐层计算结果，算是将RNN的cell转换为Transformer，但是传统的RNN每次计算一个time step，现在一个transformer内包含了多个词。
- **相对位置向量**，将原来(词向量，位置向量)的公式展开后，变为四个元素，并对其中的一些变量进行替换，消除原有的绝对位置向量部分。
- 可以应用在字符向量的训练上。
- Transformer很容易通过增加层数来提高效果。

Transformer-XL

- 虽然Transformer有很好的并行化的内力，但是self-attention的方法对



(a) Training phase. (b) Evaluation phase.

Figure 2: Illustration of the Transformer-XL model with a segment length 4.

- 可以应用在字符向量的训练上。
- Transformer很容易通过增加层数来提高效果。

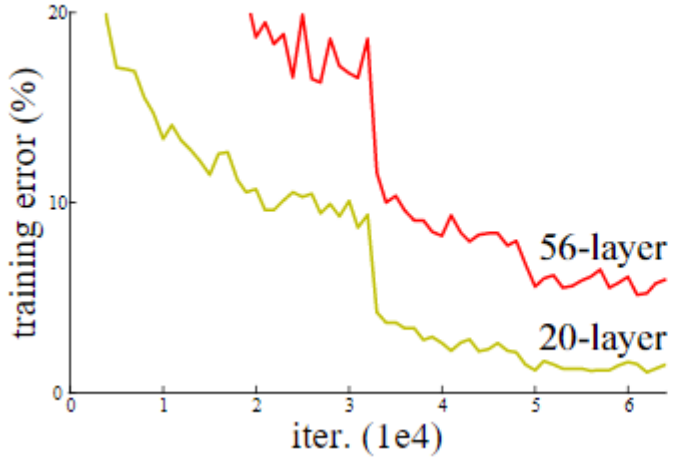
八

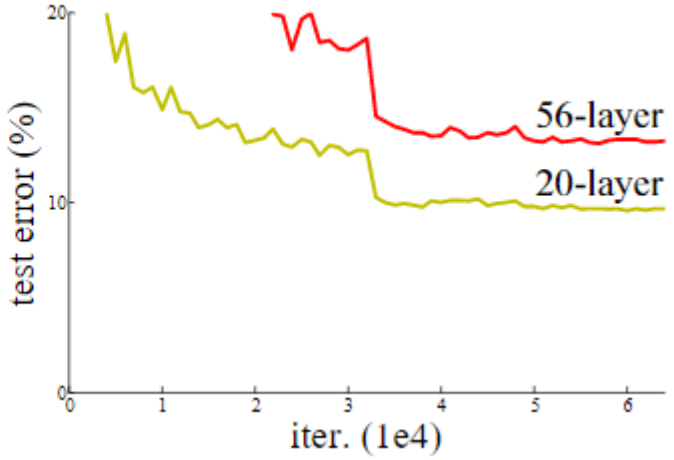
Deep Residual Learning for Image Recognition

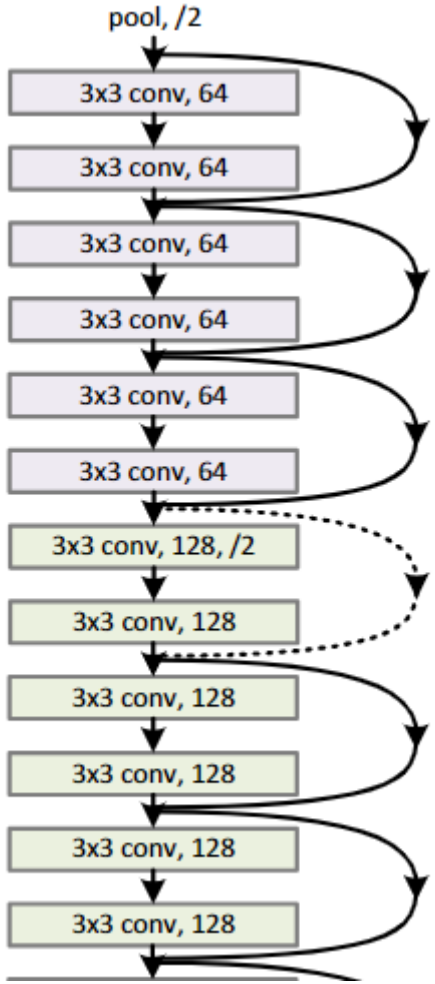
- 深层神经网络受到梯度消失和梯度爆炸的限制，无法太深。通过残差跳跃连接，可以加深网络的深度。
- 拟合残差：目标层输出与输入的差值。
- 与highway类似，但是不通过门控来控制。
- 深层神经网络训练误差和测试误差更大，证明不是由于模型复杂度过高导致的过拟合。
- 如果输入与输出的维度不同的话，通过(1)zero padding或者(2)权重矩阵变换调整维度。
- 该层实际输出：$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$

Residual Networks

- 虚线部分代表维度调整。
- 实验效果表明，可以训练到1000+层，效果仍有提升。
- 目前已经广泛应用于深度学习领域。







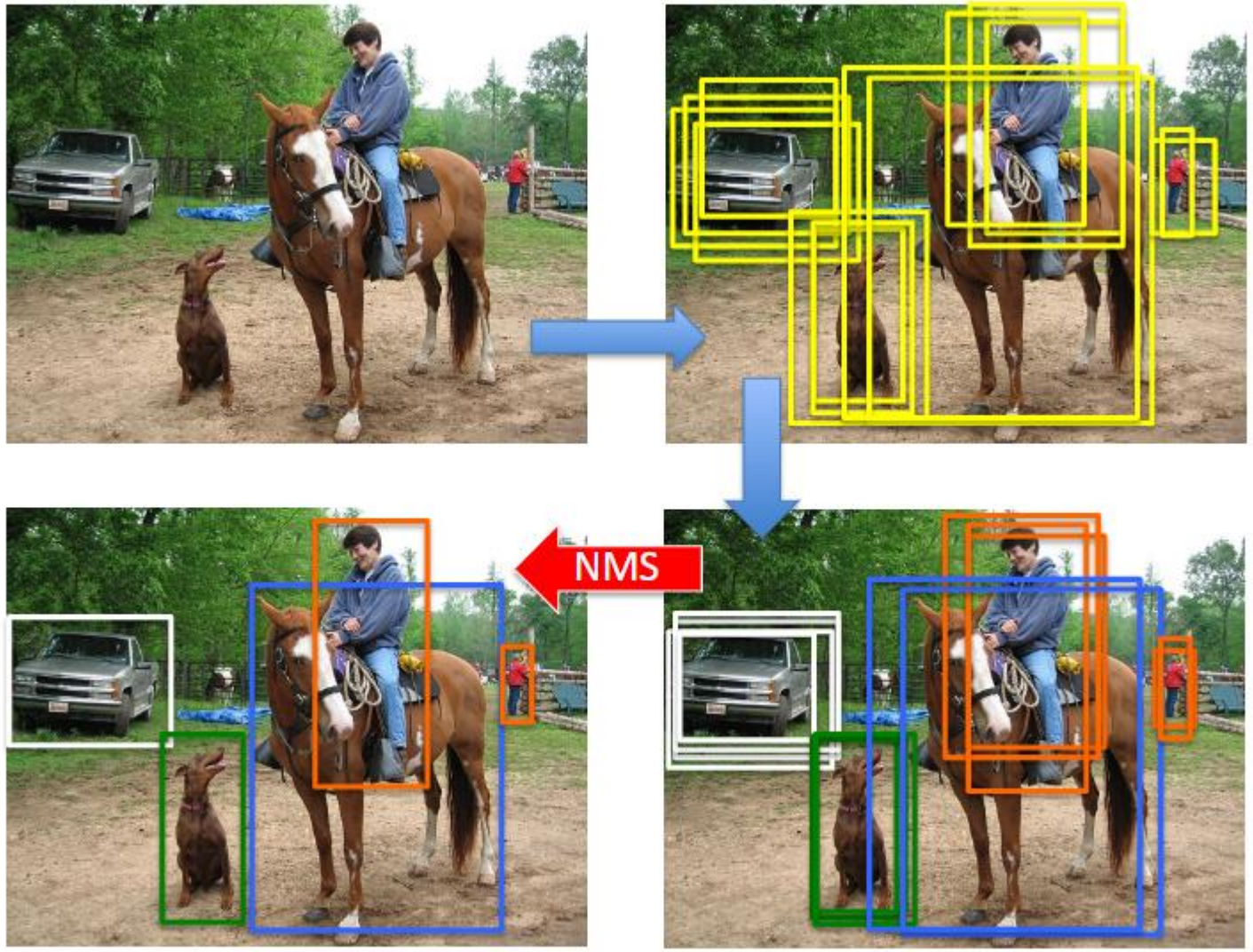
九

Soft-NMS -- Improving Object Detection With One Line of Code

- https://arxiv.org/abs/1704.04503
- NMS是Non-maximum suppression，非最大化抑制，在目标检测的过程中，重叠框的比例大于一定预设阈值的集合中，选择评分最高的。这种强制的方法可能会对重叠物体的识别效果不好，所以需要一种软的NMS。
- NMS用来减少False Positive数量，防止重复识别

Soft NMS　　Object Detection

- https://a
- NMS是N
  的过程中
  最高的。
  以需要一
- NMS用于
- 。



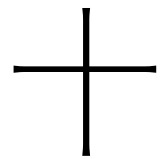
在目标检测
选择评分
是不好，所

Soft NMS —— Object Detection

- Soft NMS，在重叠高于阈值时，不是直接抛弃，而是赋予一个较低的概率分。

$$s_i = \begin{cases} s_i, & \text{iou}(\mathcal{M}, b_i) < N_t \\ s_i(1 - \text{iou}(\mathcal{M}, b_i)), & \text{iou}(\mathcal{M}, b_i) \geq N_t \end{cases},$$

- 一个简单的操作就能带来性能提升。



Watch What You Just Said:Image Captioning with Text-Conditional Attention

- 探索注意力机制在图像和文本上下文的表现，提出text-conditional attention.
- 主要流程是，根据图像特征向量和已经生成的词语来生成下一个词。
- 采用gLSTM+CNN微调的架构模型。
- 文本中可能会存在一些视觉中看不到的物体。
- Guiding LSTM采用n-gram结合图像向量来生成LSTM表示。

"After dinner, the man is comfortably lying on the sofa and watching TV."

Watch What You Just Said:Image Captioning with Text-Conditional Attention

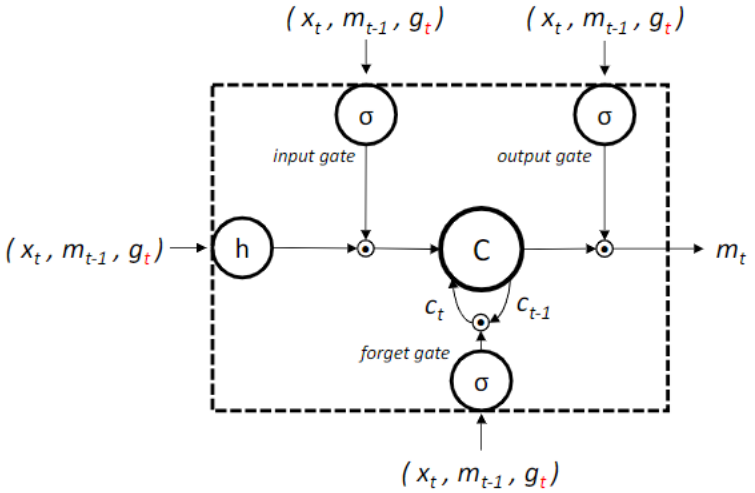
- gLSTM神经元不会迭代输入图像向量。
- Time-dependent gLSTM提供语义指导。
- 架构相比其他模型，不需要找大量图像特征，端到端。
- 图像作为语义条件，与文本进行注意力机制编码，

$$g_t^i = \sum_{j,k} W_{ijk} I^j S_t^k + b^i$$

- 为了防止参数量过大，引入转换矩阵，

$$g_t = \Phi(I \odot W_c S_t)$$

- Text-based attention.



Watch What You Just Said:Image Captioning with Text-Conditional Attention

- 利用n-gram的文本建模

$$g_t = \Phi(I \odot W_c \sum_{k=1}^{t} \frac{S_{k-1}}{t})$$

- 实验结果，n-gram比句子有更好的表现，1-gram对于注意力机制没有很好地发挥作用。
- 图像的编码层对于生成结果很重要。

Watch What You Just Said:Image Captioning with Text-Conditional Attention

Positives Examples

img-glstm: a group of people sitting around a table
NT2: group of people standing around a kitchen
sc-tanh: a group of people in a kitchen preparing food

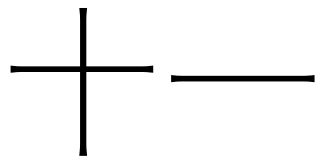
img-glstm: a man riding a snowboard down a snow covered slope
NT2: a man riding a snowboard down a snow covered slope
sc-tanh: a man flying through the air while riding a snowboard

img-glstm: a baseball player swinging a bat at a ball
NT2: a group of people playing a game of Frisbee
sc-tanh: a group of men playing a game of soccer

img-glstm: a man in a red shirt and a red fire hydrant
NT2: a little boy sitting on a skateboard on a sidewalk
sc-tanh: a young boy is holding a yellow fire hydrant

img-glstm: a herd of cattle grazing on a lush green field
NT2: a couple of horses standing on top of a hill
sc-tanh: a couple of horses are standing in a field

img-glstm: a person holding a cell phone in their hands
NT2: a person holding a pair of scissors on a table
sc-tanh: a person holding a pair of scissors in their hand

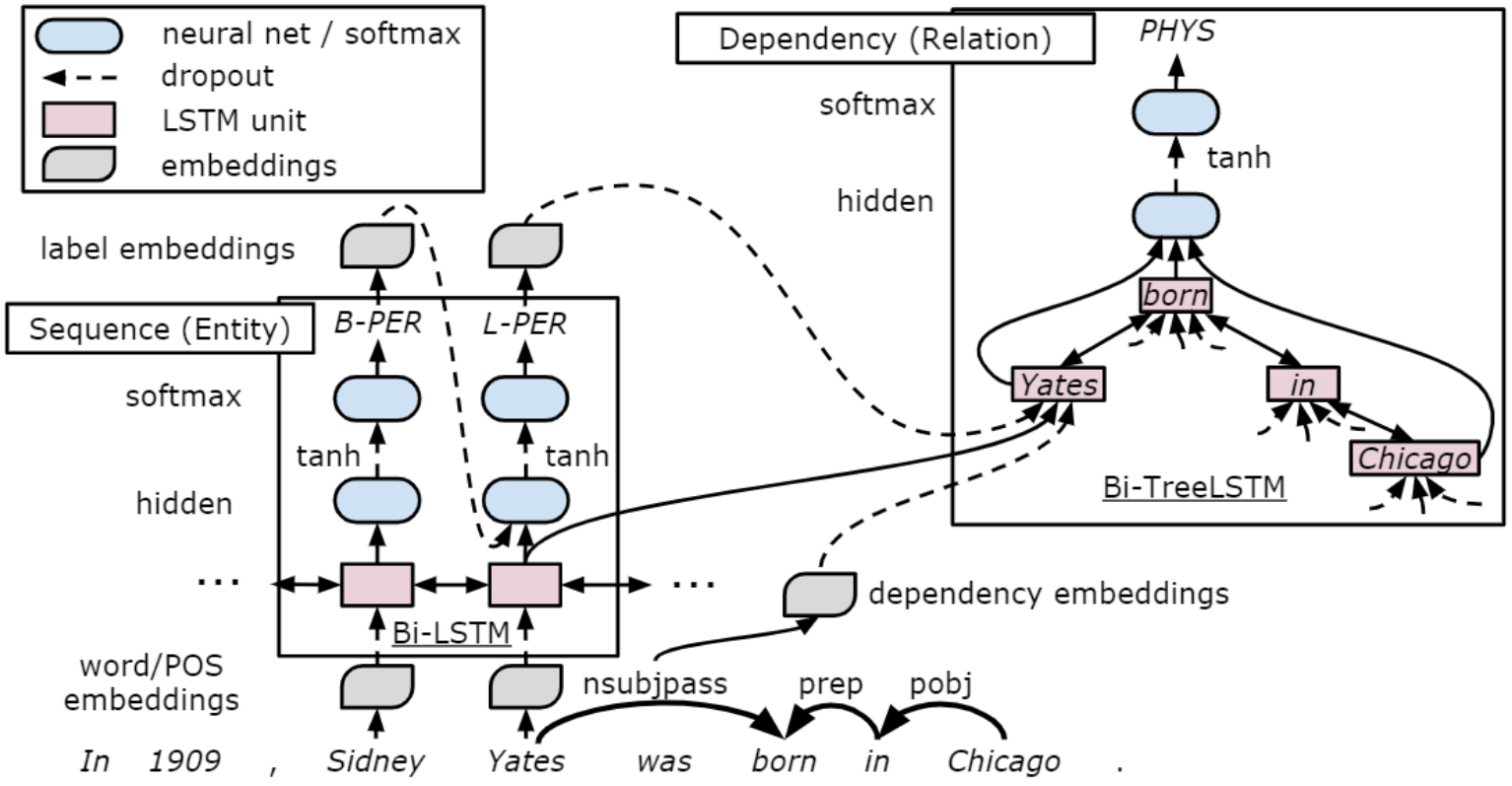


End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures

- 使用LSTM结构，同时识别出实体和关系。
- 采用双向RNN结构，输入文本后，首先检测实体。在训练过程中，采用实体预训练而不是作为真实标签来直接训练，缓解了训练早期实体识别错误率过高的问题。
- 实体的预测和依赖预测共同构成预测关系树，参数部分共享。
- 实体检测层就是一个序列分类预测。
- 在进行依赖预测的时候，选择一个没用过的节点和目标节点，寻找**最短**的路径预测。为了能够应对多个子节点，不仅是简单的节点加和，同时考虑节点的类型，同类型之间共享参数矩阵。

End-to-End Relation Extraction using LSTMson Sequences and Tree Structures

- 使用l
- 采用
  采用
  期实
- 实体l
- 实体
- 在进
  找**最**
  点加



过程中，
训练早

享。

点，寻
单的节

。

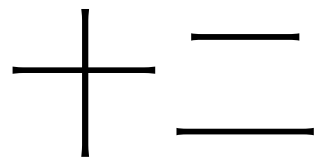
End-to-End Relation Extraction using LSTMson Sequences and Tree Structures

- 单词之间的关系预测融合了多个向量，通过softmax进行分类。
- 端到端的训练是很方便的。

Corpus	Settings	Entity			Relation		
		P	R	F1	P	R	F1
ACE05	Our Model (SPTree)	0.829	**0.839**	**0.834**	0.572	**0.540**	**0.556**
	Li and Ji (2014)	**0.852**	0.769	0.808	**0.654**	0.398	0.495
ACE04	Our Model (SPTree)	0.808	**0.829**	**0.818**	0.487	**0.481**	**0.484**
	Li and Ji (2014)	**0.835**	0.762	0.797	**0.608**	0.361	0.453

Table 1: Comparison with the state-of-the-art on the ACE05 test set and ACE04 dataset.

Settings	Entity			Relation		
	P	R	F1	P	R	F1
Our Model (SPTree)	0.815	0.821	0.818	0.506	0.529	0.518
−Entity pretraining (EP)	0.793	0.798	0.796	0.494	0.491	0.492*
−Scheduled sampling (SS)	0.812	0.818	0.815	0.522	0.490	0.505
−Label embeddings (LE)	0.811	0.821	0.816	0.512	0.499	0.505
−Shared parameters (Shared)	0.796	0.820	0.808	0.541	0.482	0.510
−EP, SS	0.781	0.804	0.792	0.509	0.479	0.494*
−EP, SS, LE, Shared	0.800	0.815	0.807	0.520	0.452	0.484**

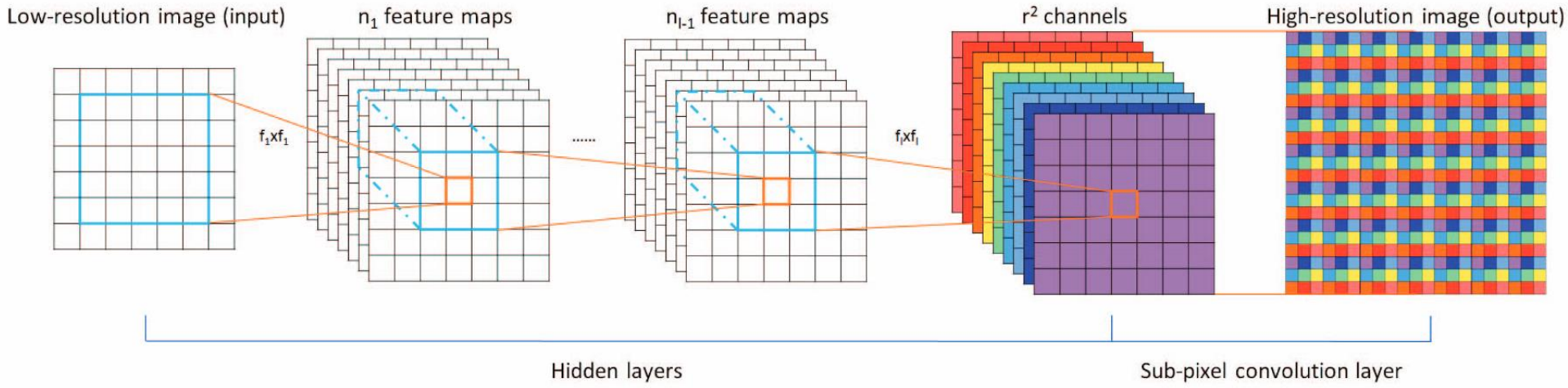


Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network

- 通过上采样的方法，减少图像在卷积神经网络重建过程中信息的损失。
- 如果对输入的图像提前采用超分辨率的算法，虽然也会有同样的效果，但是计算的复杂度却大大增加了。
- 使用K2 GPU，可以进行实时超分辨率的算法实现。
- Upscale上采样把low resolution转换到high resolution。
- 本文介绍如何有效学习upscale中的过滤器参数，取代了手工方式。
- 插值法没有解决ill-posed的问题。
- 卷积的输出比输入尺寸大的情况，很难有快速实现的解决方案。Sub-pixel在最后一层才添加超分辨率的操作。

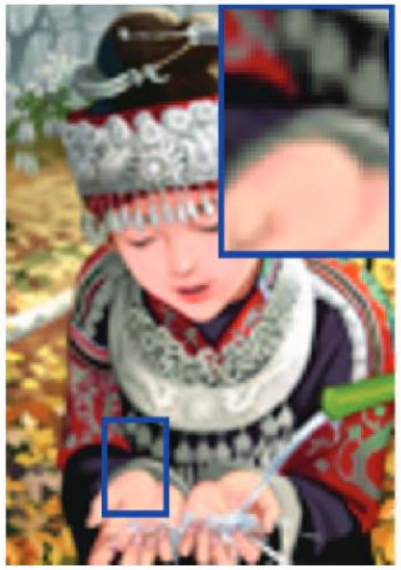
Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network

- 在后面才使用上采样，即CNN特征抽取后，减少了运算的复杂度。
- 在最后一层使用可学习的filter，而不是放在第一层的固定filter，可以有更好的性能。
- 因为具有很快的运行速度，可以在1080P的视频中实时处理。

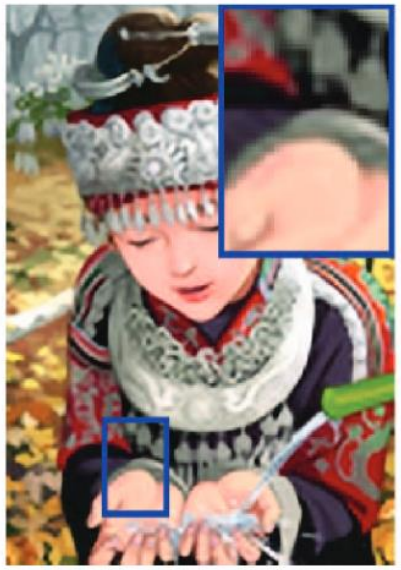


Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network

(f) Comic Original

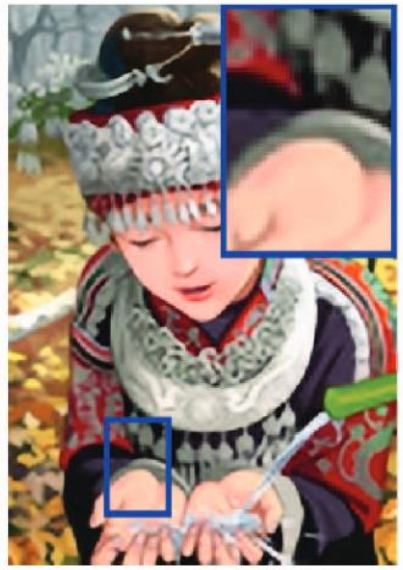


(g) Bicubic / 23.12db



(h) SRCNN [7] / 24.56db

(i) TNRD [3] / 24.68db



(j) ESPCN / **24.82db**

十三

Deep Image Prior

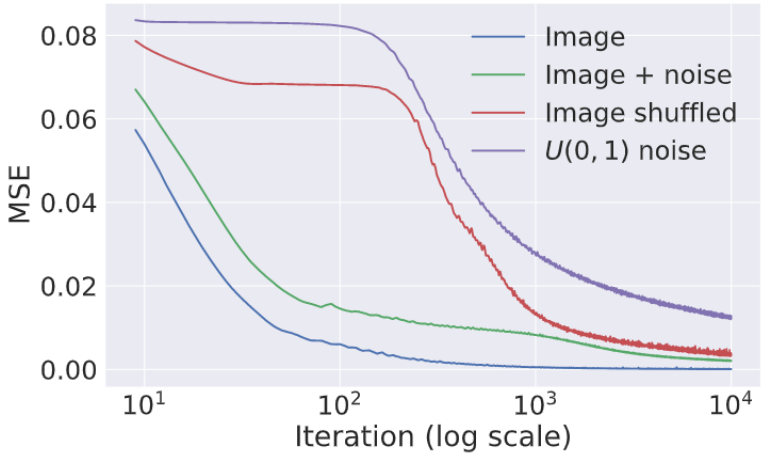
- CNN的生成模型可以学到先验知识，用来生成和改善图片。
- 先验可以通过学习，也可以通过手工设计。
- 根据数据来得到Prior，权重都是随机初始化得到的。
- 减少迭代次数，最小化映射参数θ。

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, E(f_\theta(z); x_0), \qquad x^* = f_{\theta^*}(z)\,.$$

$$E(x; x_0) = \|x - x_0\|^2$$

$$E(x; x_0) = \|d(x) - x_0\|^2$$

$$E(x; x_0) = \|(x - x_0) \odot m\|^2,$$



Deep Image Prior

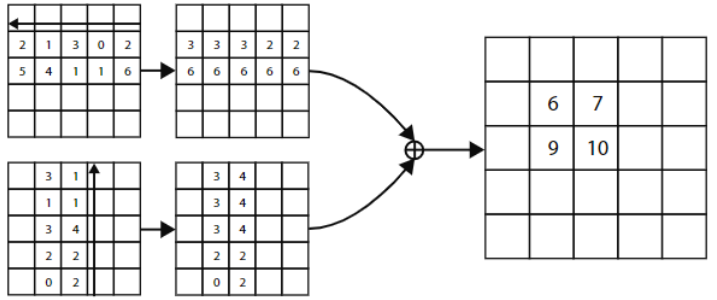
- 证明了不需要大量的训练，只需要随即初始化参数权重或者手工设计就可以很好地完成图像重构的任务。
- 相比同样非学习性的插值方法，效果更好。

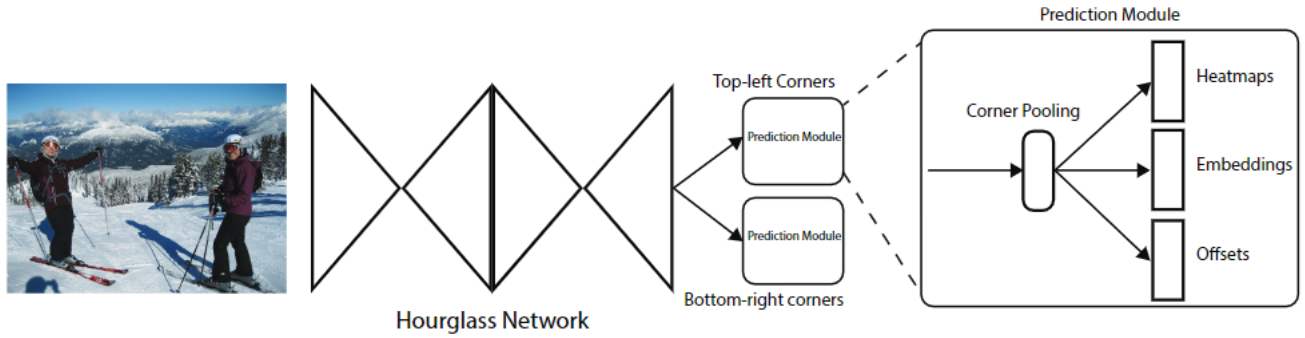
(a) Original image (b) Bicubic, **Not trained** (c) Ours, **Not trained** (d) LapSRN, **Trained** (e) SRResNet, **Trained**

十四

CornerNet: Detecting Objects as Paired Keypoints

- 单阶段的目标检测算法。
- Free-anchors.
- 定位目标的左上和右下的corner points.
- 新的训练方法。
- Corner pooling. 确定预测点的时候，找到水平和垂直的最大值点。





CornerNet: Detecting Objects as Paired Keypoints

- 当图像中有多个识别目标时，使用图像embedding聚类，用距离判断是否属于同一类。

$$L_{pull} = \frac{1}{N}\sum_{k=1}^{N}\left[(e_{t_k} - e_k)^2 + (e_{b_k} - e_k)^2\right],$$

$$L_{push} = \frac{1}{N(N-1)}\sum_{k=1}^{N}\sum_{\substack{j=1 \\ j\neq k}}^{N}\max\left(0, \Delta - |e_k - e_j|\right),$$

- 主要特征抽取网络是hourglass网络，由卷积层，池化层和上采样层组成。

十五

YOLOv3: An Incremental Improvement

- 更快，准确率更高。
- 中心点格子负责预测。
- 多标签预测，对于类别覆盖情况有所改进。
- 输出四个offset，一个是否目标预测，

80个类别输出。

- 和v2不同的是CNN网络的设计，上采样。
- K-means获取先验框，COCO数据集有九个框，对应不同尺度CNN的预测。
- 每个ground truth选择一个最大IOU框作为positive bound.

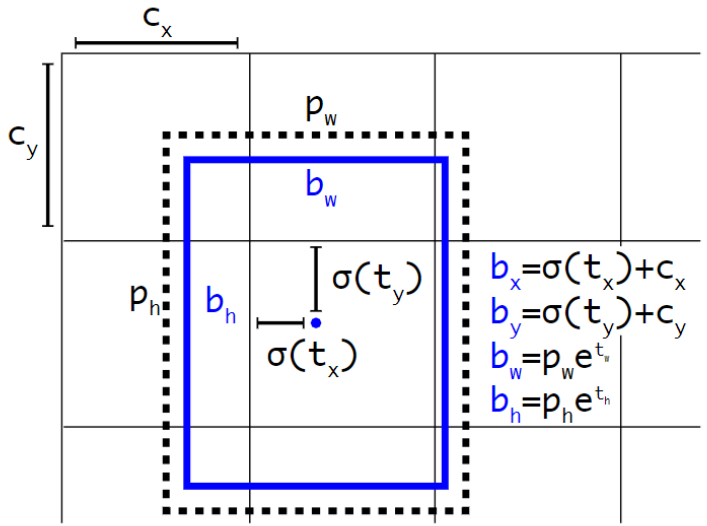


Figure 2. **Bounding boxes with dimension priors and location prediction.** We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function. This figure blatantly self-plagiarized from [15].

YOLOv3: An Incremental Improvement

- 实验结果

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

- 很快，很准，增强了对小物体的识别能力。
- Focus loss不起作用，**可能**是由于模型已经分离了objectiveness和分类预测阶段，导致大部分预测没有loss。
- 将目前目标检测的优秀手段集成到一起，速度更快，准确度更高.

YOLOv3: An Incremental Improvement

- 实验结果

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

- 很快，很准，增强了对小物体的识别能力。
- Focus loss不起作用，**可能**是由于模型已经分离了
  分类预测阶段，导致大部分预测没有loss。

起，速度更

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

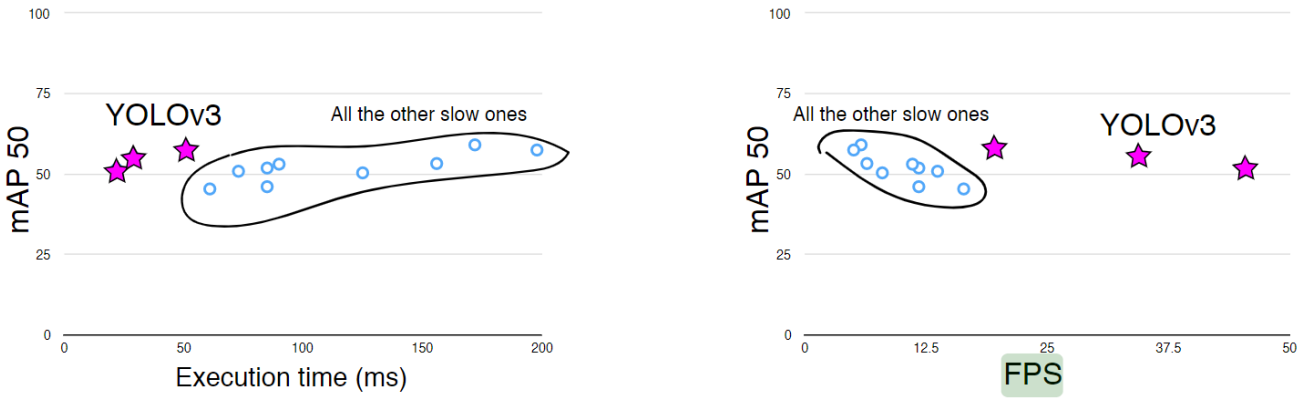


Figure 4. Zero-axis charts are probably more intellectually honest... and we can still screw with the variables to make ourselves look good!