



人工智能期中项目展示

组员：17341059黄杨峻 17341035冯家苇

题目：CNN&RNN实战





目录 | CONTENT

- 1 实现思路
- 2 网络结构
- 3 结果分析



实现思路——进度安排

第一阶段—数据处理

CIFAR-10的数据处理较为简单，torchvision调用即可，没有花费多少时间。

STSbenchmark的数据由于要做词嵌入，所以要使用到gensim的一些库函数，包括Word2Vec嵌入方法等。

第二阶段—搭建基准网络

CNN部分我们尝试了**LeNet、ResNet、VGG**等，模型的构思和搭建都较为简单，难度主要是调参

RNN部分稍为复杂，我们尝试了**单向RNN和单双向的LSTM**，以**余弦相似度**做为相似性度量函数。

第三阶段—优化创新

CNN部分，我们重新回到了数据处理部分，对CIFAR-10的载入，网络模型和损失函数部分做文章。训练效率上大大提升。

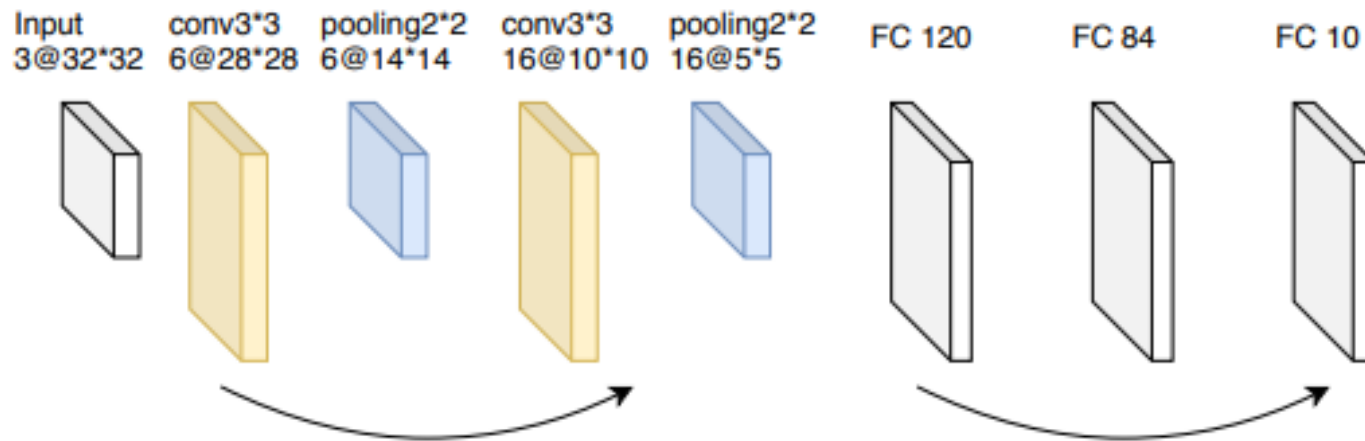
RNN部分，则是在网络结构和参数选择上进行调优，模型的效果大大提升。



目录 | CONTENT

- 1 实现思路
- 2 网络结构
- 3 结果分析

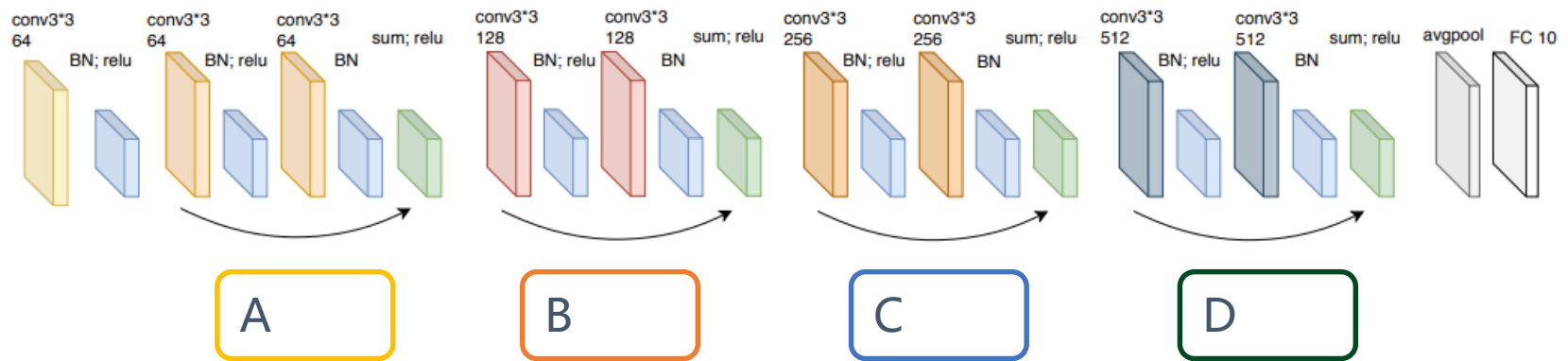
网络结构——LeNet



包含两个卷积层，两个池化层，三个全连接层，
最经典的CNN模型



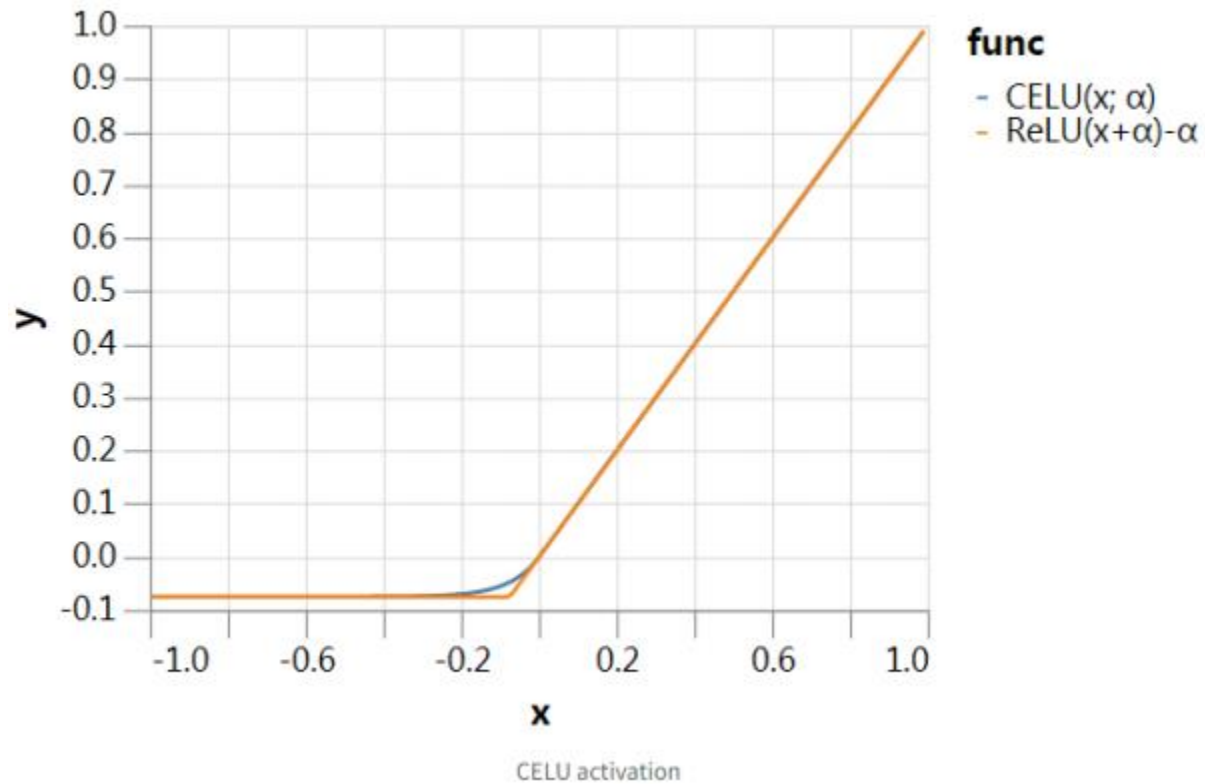
网络结构——ResNet18



首先经过一个3*3的卷积层，然后经过A,B,C,D四种block，最后经过平均池化层和全连接层。



创新——CNN：CeLU代替ReLU



创新——CNN : label smoothing



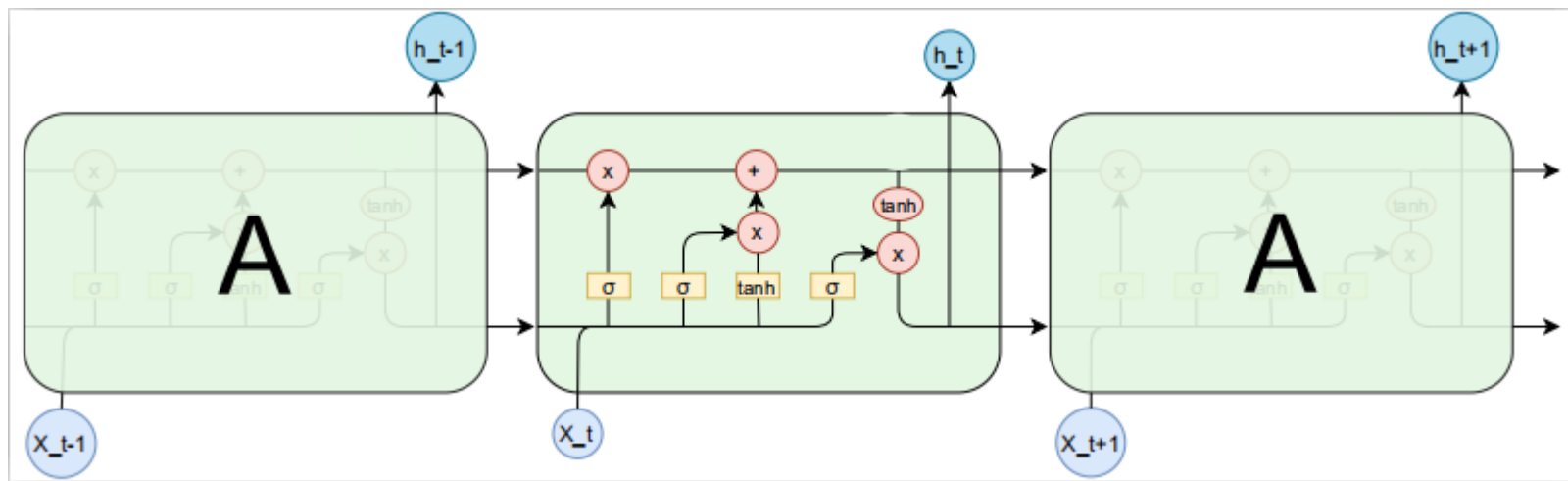
$$q'(k) = (1 - \epsilon)q(k) + \epsilon u(k)$$

$$H(q', p) = - \sum q'(k) \log p(k)$$

$$H(q', p) = (1 - \epsilon)H(q, p) + \epsilon H(u, p)$$

标签平滑是一种在分类问题中提高神经网络训练速度和泛化能力的成熟技巧。它包括混合独热目标概率与均匀分布在类标签内的交叉熵损失。这有助于稳定的产生分布，并防止网络做出过度自信的预测，而这种预测可能会抑制进一步的训练。

网络结构——LSTM



$$f_t = \sigma(W_f * \text{concat}(h_{t-1}, x_t) + b_f)$$

$$i_t = \sigma(W_i * \text{concat}(h_{t-1}, x_t) + b_i)$$

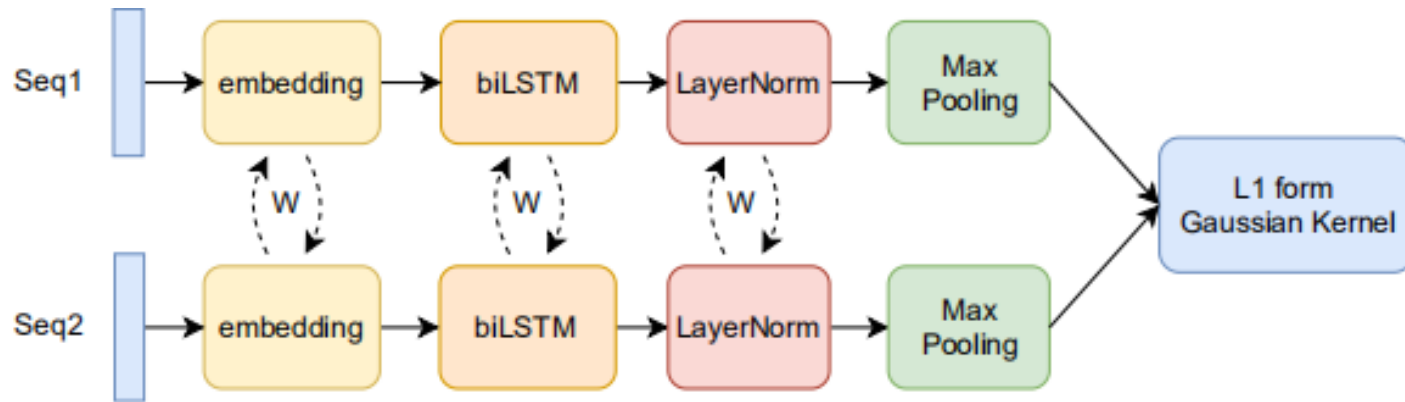
$$\hat{C}_t = \tanh(W_C * \text{concat}(h_{t-1}, x_t) + b_C)$$

$$o_t = \sigma(W_o * \text{concat}(h_{t-1}, x_t) + b_o)$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t$$

$$h_t = o_t * \tanh(C_t)$$

网络结构——Siamese-biLSTM-NP



两个序列分别训练，每个网络包含词嵌入，双向LSTM，正则化，池化层三个函数，最后通过一范数高斯核作为相似度度量指标。

$$g(h_{T_a}^a, h_{T_b}^b) = \exp(-\|h_{T_a}^a - h_{T_b}^b\|_1)$$

$$\mathbf{a}^t = W_{hh} h^{t-1} + W_{xh} \mathbf{x}^t$$

$$\mathbf{h}^t = f\left(\frac{\mathbf{g}}{\sqrt{(\sigma^t)^2 + \epsilon}} \odot (\mathbf{a}^t - \mu^t) + \mathbf{b}\right) \quad \mu^t = \frac{1}{H} \sum_{i=1}^H a_i^t \quad \sigma^t = \sqrt{\frac{1}{H} \sum_{i=1}^H (a_i^t - \mu^t)^2}$$

其他



end-to-end model

Pretrain

Pytorch.lr_scheduler

Label 调整

padding 方法与embedding 矩阵的调整



目录 | CONTENT

- 1 实验思路
- 2 网络结构
- 3 结果分析



结果分析——CNN

表 1: LeNet:lr = 0.01, weight_decay=5e-4, epoch=80

Class	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
Accuracy	73%	93%	88%	47%	71%	50%	67%	92%	91%	95%
Train accuracy	74.2%									
Valid accuracy	71.7%									

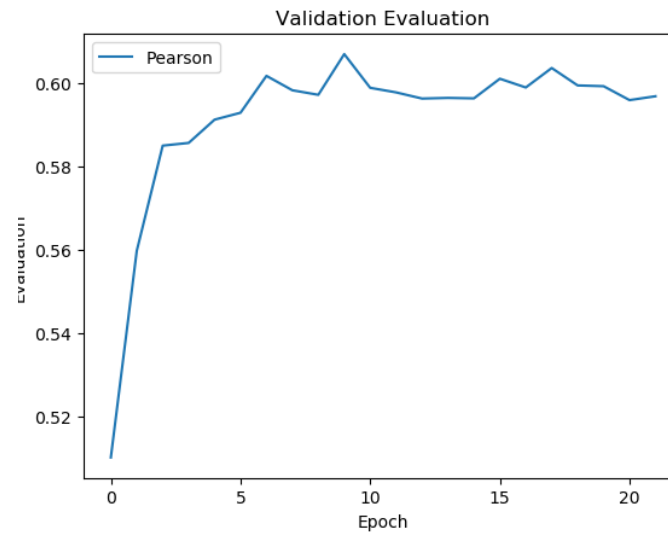
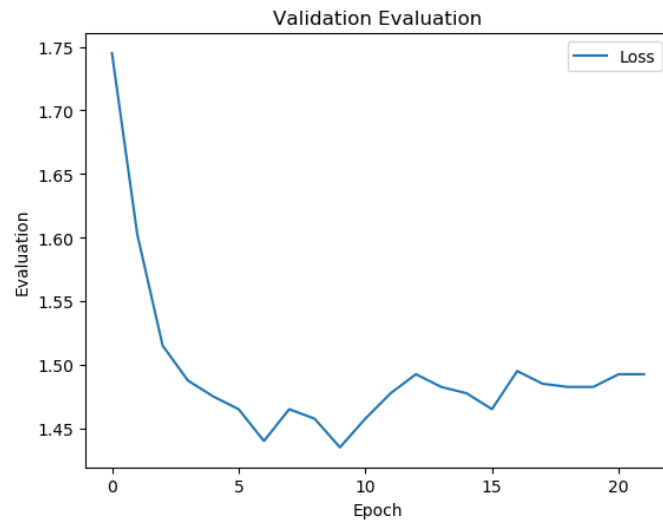
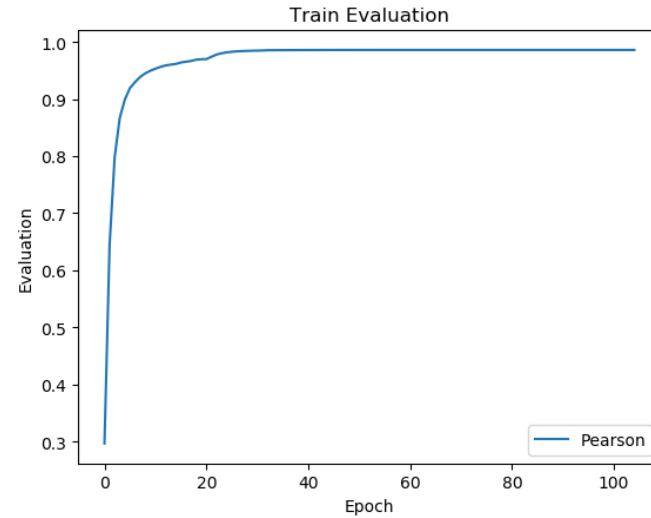
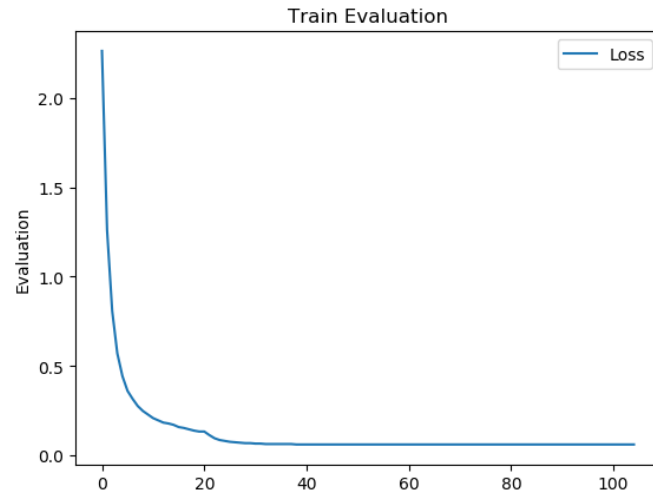
表 2: ResNet:lr = 0.01, weight_decay=5e-4, epoch=30

Class	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
Accuracy	99%	97%	95%	88%	90%	82%	84%	98%	96%	98%
Train accuracy	95.6%									
Valid accuracy	88.9%									

结果表明，ResNet效果明显优于LeNet



结果分析——RNN





结果分析——RNN

表 4: Ablation Study

model	Test Loss	Test Pearson
LSTM(1 layer, fixed embedding)	3.05	0.37
LSTM(2 layers, fixed embedding)	3.02	0.38
biLSTM(1 layer)	2.99	0.38
biLSTM(1 layer, LN)	2.33	0.48
biLSTM(1 layer, GCN via time-step)	2.54	0.46
biLSTM(1 layer, LN, MP)(ours)	1.52	0.57

Reference



- [1] Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, Christian Szegedy, Vincent Vanhoucke. Rethinking the inception architecture for computer vision. <https://arxiv.org/abs/1512.00567>, 2015.
- [2] Arizona Phoenix. Siamese recurrent architectures for learning sentence similarity. <https://dl.acm.org/citation.cfm?id=3016291>, 2016.
- [3] pytorch 官方文档. <https://pytorch.org/docs/stable/index.html>.
- [4] Yann LeCun, S. Chopra, R. Hadsell. Learning a similarity metric discriminatively, with application to face verification. <http://yann.lecun.com/exdb/publis/pdf/chopra-05.pdf>, 2005.
- [5] Mona Diab, Daniel Cera. Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. <https://www.aclweb.org/anthology/S17-2001.pdf>, 2017.